MULTILEVEL UPSCALING AND VALIDATION FRAMEWORK FOR SIMULATION OF TURBINE ENGINE COMPONENTS

A Dissertation
Presented to
The Academic Faculty

by

Austin McKeand

In Partial Fulfillment
of the Requirements for the Degree
Mechanical Engineering in the
Woodruff School of Mechanical Engineering

Georgia Institute of Technology
July 2019

MULTILEVEL UPSCALING AND VALIDATION FRAMEWORK FOR SIMULATION OF TURBINE ENGINE COMPONENTS

Approved by:

Dr. Seung-Kyum Choi, Advisor
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Yan Wang
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Roger Jiao
School of Mechanical Engineering
Georgia Institute of Technology

Date Approved: July 17, 2019

ACKNOWLEDGEMENTS

Primarily, I would like to thank my advisor, Dr. Seung-Kyum Choi, for helping me through these two years by consistently encouraging me to work diligently to obtain good results as well as produce high quality presentations, written reports, and journal articles. Thanks to his advisement and experience, I have improved as a researcher as well as learned many new skills along the way. I would also like to thank my reading committee members, Dr. Yan Wang and Dr. Roger Jiao, for taking the time to examine my thesis and suggest edits.

In addition, this thesis would not have been possible without my friends. They have been there to support me through all of the tough weeks and many hours that this research has required. Their love and support has made the worst weeks bearable and the best weeks restful.

Lastly I would like to thank my parents. They have been supportive of me through everything and are always willing to listen. Their advice and wisdom has helped me navigate life before and during my college career.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

A	covariance, cross-sectional area
b	PCE coefficient
B	Bernstein polynomial
cov	covariance
d	sum of squared differences
E	elastic modulus
F	cumulative distribution function, force
f	arbitrary black box function
h	lag distance
m	number of pairs
M	statistical moment
N	number of samples
P	eigenvector matrix, control points
TF	transfer function
u	random variable
w	random field
W	covariance matrix
x	random variable
Y	random output
α	significance level
δ	strain
ε	error
η	input to random field

θ	input to random variable
λ	semivariogram value
λ	eigenvalue
Λ	eigenvalue matrix
ξ	standard uncorrelated random variable
σ	stress, standard deviation
Φ	eigenfunction
ϕ	Askey-chaos orthogonal polynomial

SUMMARY

Efficient modeling of manufacturing uncertainty is critical in the design of turbine engine components. In this research, a multilevel validation framework is developed to efficiently account for the geometric and material uncertainty associated with the manufacturing process to accurately predict the performance of engine components. This framework is created to handle both epistemic and aleatory uncertainty. Specifically, the spatial variability of the uncertain geometric parameters obtained from coordinate measuring machine data of manufactured parts is represented as aleatory uncertainty. Porosity and defects in the manufactured parts based on 3D X-ray CT scanned images are represented as epistemic uncertainty. Multiple efficient statistic tools are integrated into the proposed framework. Karhunen-Loeve expansion is utilized to create a set of correlated random variables from the obtained uncertainty data and a fine scale finite element model of the component is created that accounts for the uncertainties quantified by these correlated random variables. A stochastic upscaling method is then developed to form a simplified model that can represent this detailed model with high accuracy under uncertainties. In addition, a validation method for multi-variate responses is developed and used to validate the simulation results with the experimental results. The modal frequency analysis of a turbine blade example is used to demonstrate the efficacy of the proposed framework. The application results show that the proposed method effectively captures the geometric uncertainties introduced by manufacturing while providing accurate predictions under uncertainties.

CHAPTER 1. INTRODUCTION

As advancements in simulation technology have improved, the desire to have simulations more accurately represent reality have grown. Engineering analysis typically assumes that material properties and loading/boundary conditions are constant and homogenous. In addition, the manufactured part is typically assumed to have the same geometry as the CAD part it was made from. However, these two assumptions are rarely the case in reality. Microdefects, such as cracks or grain boundaries in materials, as well as porosities introduced during manufacturing can cause it to have non-homogenous properties. Manufacturing techniques are never perfect as well, so there is also usually slight deviation between the manufactured part and CAD model of the part.

When performing engineering analysis of turbine blades, it is extremely important to have accurate results. Small deviations in properties can cause catastrophic failure in the turbine engines. In addition, because of the extreme conditions that turbine blades operate in, even slight differences in geometry can cause massive variations in the performance of the turbine blade, changing various failure states such as maximum stress, maximum tip deflection, and natural frequency. Because of this incorporating the uncertainties associated with the manufacturing of turbine blades into simulation is extremely important. This thesis presents a framework that utilizes stochastic upscaling to incorporate uncertainty quantification into simulation of turbine blades. In addition, validation of the results will be introduced to ensure the model is a good representation of reality.

The organization of the introduction will be as follows. Sections 1.1 and 1.2 will discuss a background of turbine engines and uncertainty quantification, respectively.

Section 1.3 will discuss a background of the upscaling process and its existing uses in engineering analysis. Section 1.4 will discuss a background of verification and validation and their usefulness to simulation. Section 1.5 will discuss the specific research goals for this project. Finally, Section 1.6 will outline the organization of the chapters in this paper.

1.1 Turbine Engines

Turbine engines typically operate in extreme condition, with temperatures up to 1250 °C and speeds up to 13,000 rpm. Investigating the failure modes of turbine blades becomes extremely difficult, since small changes in properties can cause large differences in mechanical performance. Besides striking debris, blade failures are most often caused by fatigue or creep rupture associated with manufacturing defects within the blade [1]. Many examples have been done, but Mazur et al. investigated the failure of a specific gas turbine blade mad of Inconel 738LC alloy, and concluded that the combination of operating in extremely high temperatures, as well crack propagation from a grain boundary defect caused the failure of the blade [2].

In general crack propagation often causes failure in turbine blades. Although performing stress analysis or displacement analysis to investigate stress or strain failures can be useful for determining possible failure modes, more often than not it is vibration that causes microdefects in the blade to propagate into large enough cracks that failure can occur. Because of this, investigating the natural frequency and modal shapes of turbine blades is extremely important. Hou et al. did a thorough investigation of fatigue failures using finite element analysis and found that blades being excited at their natural frequency

can often cause crack propagation. Accurate simulation to determine these possible natural frequencies has become increasingly important [1].

In addition, using stochastic simulation for finite element analysis of turbine blades has been done before. Garzon and Darmofal [3] used principal component analysis (PCA) to analyze uncertainty of the geometry caused by manufacturing defects. Carnevale et al. [4] modeled uncertainties with a Gaussian distribution. In addition, Javed et al. [5] used Monte Carlo sampling to help quantify geometric uncertainty in blade simulation. Lange et al. [6] also generated blade models based on geometric uncertainties, but used Gaussian distributions found from measured geometric data. Thakur et al. [7] used a PCA-based method to quantify the spatial variance of geometric parameters. For material properties, Nagpal et al. [8] used uncorrelated random variables based on Gaussian distributions to change the elastic modulus of turbine blades in simulation, however these distributions were chosen by the researchers based on past experience. Zhu et al. [9] performed similar simulations but attempted to represent the distribution based on experimental results. Weiss et al. and Shen similarly implement material property uncertainty into simulations in order to more accurately predict the performance of turbine blades [10][11].

Using uncertainty quantification to predict the performance of turbine blades has become increasingly important, but exactly how to implement these uncertainties into simulation has been changing over time. In the past, many researchers would use their own experience to modify parameters in simulation, but more recently efforts have been made to justify these modification with experimental results. This project will attempt to use experimental data along with several uncertainty quantification techniques to accurately implement geometric and material uncertainty into simulation.

1.2 Uncertainty Quantification

Uncertainty is often broken down into aleatory and epistemic uncertainty. Aleatory uncertainty is typically what is associated with uncertainty. Aleatory uncertainty is also known as irreducible uncertainty, and is typically described by a probability distribution. Epistemic uncertainty is used to describe a situation where there is not enough collected data to accurately describe the distribution. If enough data were collected for the epistemic variable, it would eventually become aleatory. Because an exact distribution for epistemic uncertainty cannot be obtained, it is often represented by an interval. There are other ways to represent epistemic uncertainty, but in this thesis it will always be represented by an interval [12].

Typically, in engineering analysis, some input variable are determined to be aleatory while others remain epistemic. In this case, the output contains a mix of both aleatory and epistemic uncertainty. In these cases the output is often represented using a P-box. A P-box is simply two cumulative distribution functions (CDF) which represent a left and right side boundary. The true distribution of the output is then determined to be somewhere inside of this interval. In this way, a general knowledge about the shape of the distribution can be used while still keeping the interval information from the epistemic uncertainty. Examples of aleatory, epistemic, and P-box representation of uncertainty can be seen in Figure 1-1 [12][16].

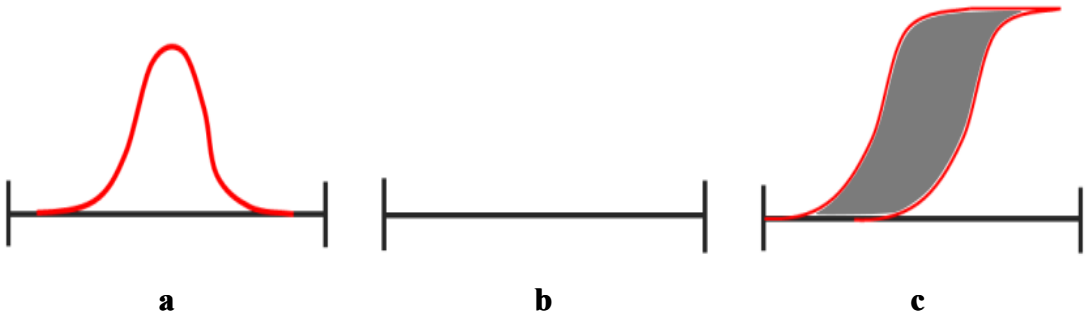


Figure 1-1: Uncertainty Representations - a) Aleatory Uncertainty b) Epistemic Uncertainty c) P-Box Uncertainty

Epistemic and aleatory uncertainties have attempted integrations into a single simulation in previous research. Ross et al. [13] integrated epistemic and aleatory uncertainty into groundwater flow and transport simulations. Hofer et al. [14] and Oberkampf et al. [15] attempted to apply epistemic and aleatory uncertainty in a general manner.

In this research, in order to obtain a p-box representation, model simulations must be conducted in a specific way. The process begins by choosing a specific set of epistemic inputs for the system. Then simulations are conducted repeatedly with the same epistemic inputs but varying aleatory inputs. In this way, a single CDF can be obtained. This process is then repeated with different epistemic inputs until several CDF's are obtained. The p-box is then taken to be the left and right bounds of all of the CDF's [16][17].

1.3 Upscaling

Upscaling is a process often used in engineering to replace detailed models with more continuous, homogenized models. The goal is to create a simple model that creates

the same results as a more fine scale, detailed model. This has been used in the past for finite element models where the material in question has complex microstructure behavior. By creating a detailed model containing the microstructure information and performing an upscaling process, homogenous material properties (such as elastic modulus, strain energy, and poisson ratio) can be determined without losing the information contained in the microstructure behavior. Ynvonnet et al. developed a homogenization technique designed for materials with heterogeneous properties such as biological parts, rubbers, and elastics [18]. This technique of upscaling/homogenization is important when considering multiscale models [19][20][21][22].

1.4 Verification and Validation

Verification and validation have become increasingly important with the improvement of simulations. Verification is essentially the process of ensuring a simulation is congruent with the related contemporary mathematical equations, and validation is the process of ensuring that a simulations correctly matches reality [23]. Due to advances in technology that allow for abundant computing power, the verification process has become easily achievable by using computers for the mathematical equations. However, validation is a more difficult process due to having to accurately replicate the boundary conditions of simulations and needing several samples to ensure a good result. In addition, the process of comparing the simulation to reality can be difficult, since knowing whether or not the results are close enough can seem arbitrary [24]. In order to overcome this difficulty, many different validation metrics have been proposed to quantify the relationship between simulation and experimental results. Many of these methods are used for validation of probabilistic responses, and Liu et al. have reviewed many of the different validation

metrics used for this process [25]. In order to accurately identify whether or not the simulations created by this framework match reality, a validation process will be implemented in the end of this framework.

1.5 Research Goals

The goal of this research is to create a framework that can accurately propagate geometric and material defect uncertainty into simulation and validate the results with experimental data. An overview of this framework can be seen in Figure 1-2.

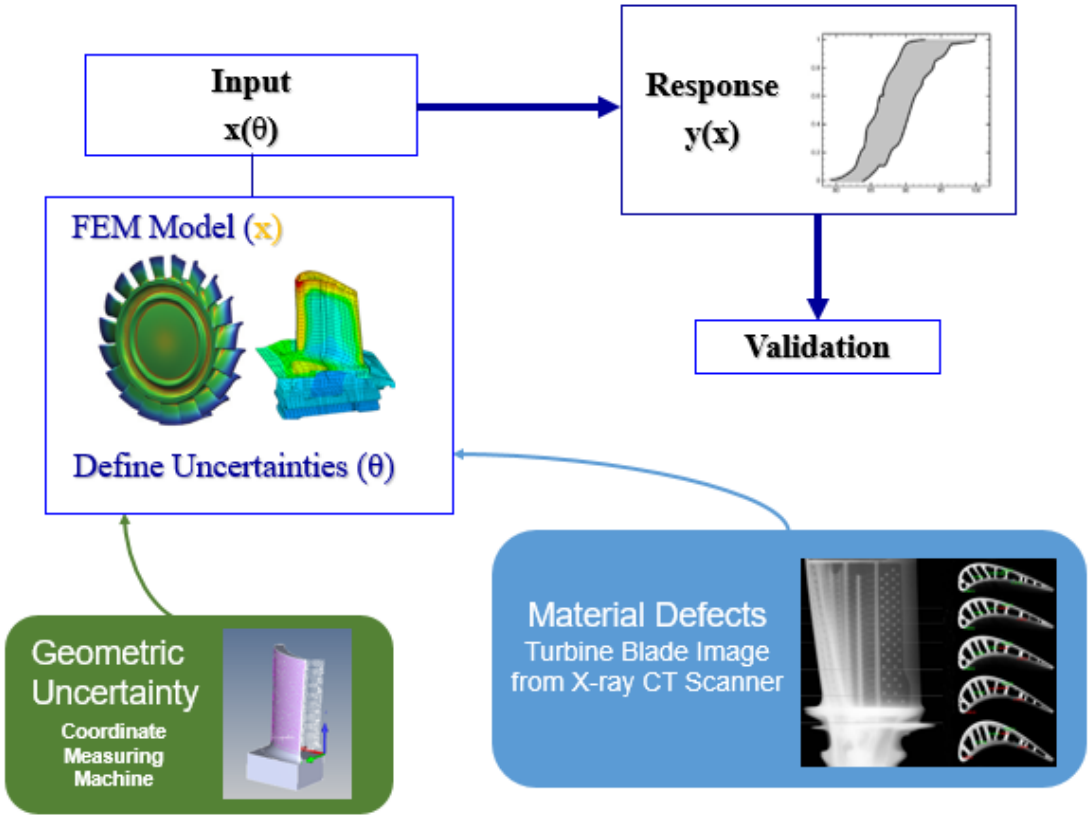


Figure 1-2: Overview of Stochastic Upscaling and Validation Framework

This process will be separated into 4 different sections. First, geometric uncertainty will be quantified using a CMM machine, and integrated into simulation. Second, material defect uncertainty will be quantified using a CT scanning machine, and upscaling will be used with simulations to obtain a homogenized elastic modulus. Third, both material defect uncertainty and geometric uncertainty will be propagated into the same model. Finally, the final obtained model will be validated using validation metrics.

1.6 Thesis Organization

The organization of this thesis will be as follows. Chapter 2 will present a background of the various methods that are used in this framework. Chapter 3 will go into detail about the geometric uncertainty quantification and application to simulation. Chapter 4 will discuss material defect uncertainty quantification and its application into simulation, as well as integrating both geometric uncertainty quantification and material defect uncertainty quantification into the same simulation. Chapter 5 will show the efficacy of the proposed upscaling and validation framework through a turbine blade example. Chapter 6 will present the conclusions and possible future work for this project. The framework that was created is applied to a turbine blade example, but is theoretically applicable to any part. Therefore, the presentation of the framework in chapters 3-4 will be in relation to a general part and simulation model.

CHAPTER 2. BACKGROUND OF METHODS

In this section, an overview and background of the methods that will be used in this framework will be discussed. Some changes have been made to some of the methods, but the focus of this section will be reviewing the existing resources on the methods.

2.1 Sampling Methods

In uncertainty quantification, there have been several methods developed in order to accomplish the task of sampling from a distribution. The two methods used in this frame work are Monte Carlo Sampling (MCS) and Latin Hypercube Sampling (LHS). Given a cumulative distribution function (CDF) represented by $F(x)$, these sampling methods randomly generate a value x.

2.1.1 Monte Carlo Sampling

MCS is a method that makes use of the inverse cumulative distribution function shown in equation 1,

$$x = F^{-1}(u) \tag{1}$$

where F^{-1} represents the inverse CDF, x represents a value of the random variable defined by the distribution, and u represents the probability of that value existing. Since the probability can only be from 0 to 1, u must exist in the range [0, 1]. MCS begins by generating a random in that range of 0 to 1 and plugging the value into the inverse CDF to obtain a value for x. This can be repeated multiple times until a suitable number of values

of *x* have been generated. The largest downside to MCS is that when relatively few values of *x* are required, MCS can cause a skewed representation of variable [12].

2.1.2 Latin Hypercube Sampling

LHS works in a similar way, using the inverse CDF to find values for *x*, however it tries to compensate for the weakness in MCS. For *n* values of *x*, the interval of 0 to 1 is subdivided into *n* different groups of the same sizes. A value is then randomly chosen in each interval, and those values are plugged into the inverse CDF function to obtain *n* values of *x*. Because the samples are generated from subdivided groups, LHS is more likely to generate a fairly even spread of variables even when only a small number are generated [12].

2.2 Semivariogram

Semivariogram (sometimes simply called variogram analysis) is the statistical process of quantifying spatial covariance of a random variable. It has been developed primarily for geostatistics [26], but is often used to help quantify the correlation between any variable that is spatially correlated. The semivariogram is calculated using equation 2,

$$\hat{\gamma} = \frac{1}{2m(h)} \sum_{i=1}^{m(h)} [z(x_i) - z(x_i + h)]^2 \tag{2}$$

where *h* represents the distance between two sample points (known as the lag distance), *m(h)* represents the number of possible pairs given lag distance *h*, x_i represents the special

location, z represents the random variable value at that location, and $\hat{\gamma}$ represents the semi-variogram value. An example of a semivariogram can be seen in Figure 2-1.

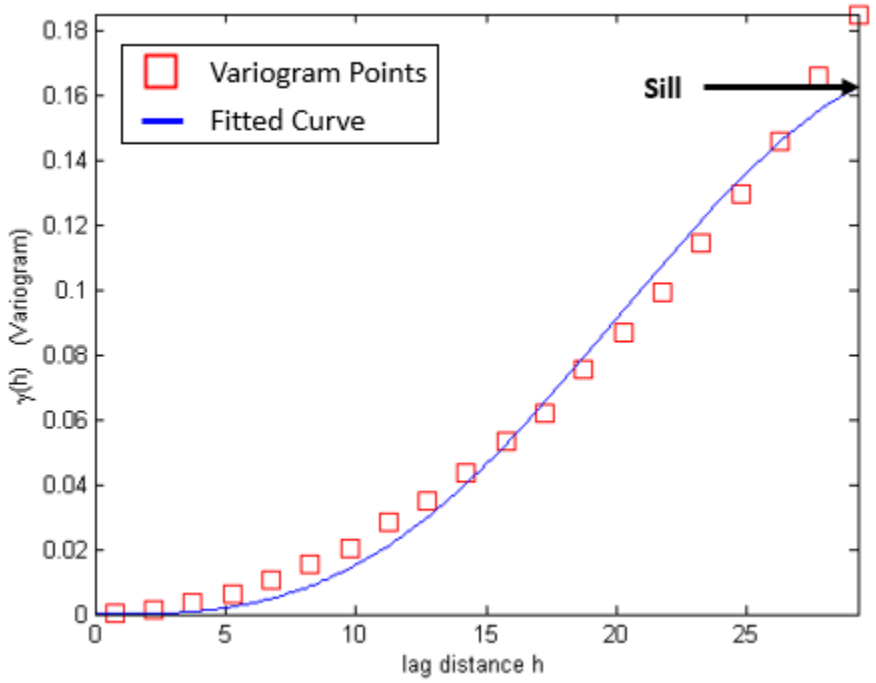


Figure 2-1: Example Variogram

After computing the semivariogram of the data, a curve is fitted in order to calculate the semivariogram value at an arbitrary distance, h. Many different types of functions can be used to fit the data, but the coefficients will be determined using least squares minimization method. Given an arbitrary function *f(h),* the sum of the squared differences is given by equation 3.

$$d = \sum_{i=1}^{n} [\hat{\gamma}(h_i) - f(h_i)]^2 \tag{3}$$

where d represents the sum of the squared differences, and n represents the number of different lag distances, h. The best fit is determined by minimizing d [27].

The semivariogram value, $\hat{\gamma}$, is closely related to the covariance of two points at lag distance, h. The covariance of two points at lag distance, h, can be calculated using equation 4.

$$cov(h) = Sill - \hat{\gamma} \tag{4}$$

where the sill represents the highest point in the semivariogram.

2.3 Karhunen-Loeve Expansion

Often times representing a random field can computationally expensive due to the large number of random variables. Karhunen-Loeve (K-L) expansion is a method used represent a random field with decreased dimensionality. K-L expansion has several uses, but in this project it is used to transform a set of uncorrelated random variables into a set of correlated random variables [10]. K-L expansion works by performing an eigenvalue decomposition of the autocovariance function, as shown in equation 5

$$w(x,\theta) = \widetilde{w}(x) + \sum_{i=1}^{\infty} \sqrt{\lambda_i}\phi_i(x)\xi_i(\theta) \tag{5}$$

where $\widetilde{w}(x)$ is the mean function of the random field, $w(x,\theta)$, $\xi(\theta)$ is the set of standard uncorrelated random variables, and λ_i and $\phi_i(x)$ are the i^{th} eigenvalues and eigenfunctions of the autocavariance function, respectively [28][29].

When the random field is discretized, the eigenvalue problem can be written in matrix form, represented by equation 6

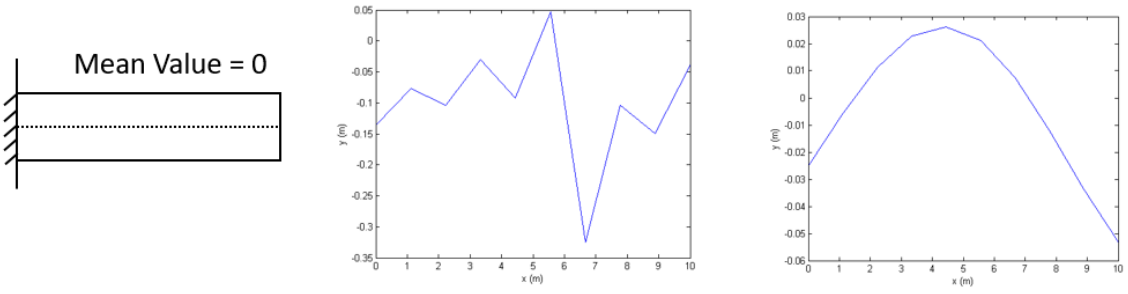
$$[P][\Lambda] = [W][P] \tag{6}$$

where $[W]$ represents the covariance matrix associated with the discretized random field, $[P]$ is the orthogonal eigenvector matric, and $[\Lambda]$ is the orthogonal eigenvalue matrix. Rearranging equation 6 results in the eigenvalue decomposition of the covariance matrix, as shown in equation 7

$$[W] = [P][\Lambda][P]^T \tag{7}$$

By only keeping the eigenvectors associated with small eigenvalues, the small contributions can be ignored and the dimensionality of the random field representation can be reduced.

An example of K-L expansion can be seen on a simple beam with varying geometry. Theoretically the beam should be a straight line, i.e. it's y-position should be 0 across the whole beam, as shown in Figure 2-2a. However in reality, the position varies slightly across the entire beam. Simply randomizing the position of the beam results in a very erratic distribution as shown in Figure 2-2b, which is fairly unrealistic. In reality it would be expected that the position of the beam would be random but smooth.



a **b** **c**

Figure 2-2: K-L Expansion Example - a) Beam Figure b) Random Deviations c) Deviations via K-L Expansion

By introducing a covariance matrix (where 2 points close to each other have high covariance and 2 points far away from each other have low covariance) and performing K-L expansion, the same set of uncorrelated random variations used to create Figure 2-2b can be transformed into a correlated set of random variables. These can be seen in Figure 2-2c with varying levels of covariance. As the covariance increases, the position of the beam becomes smoother and less erratic, but remains completely random.

2.4 Polynomial Chaos Expansion

Polynomial chaos expansion (PCE) is a stochastic expansion method used to represent a random variable through polynomials. This is done by using a set of constants (coefficients) along with a set of orthogonal polynomials to transform the standard uncorrelated random variable set into an arbitrary distribution [30]. This process is shown in equation 8

$$X(\eta) = b_0\Phi_0 + \sum_{i=1}^{\infty} b_{i_1}\Phi_1\left(\xi_{i_1}(\eta)\right) + \tag{8}$$

$$\sum_{i_1=1}^{\infty}\sum_{i_2=1}^{i_1} b_{i_1 i_2}\Phi_2(\xi_{i_1}(\eta), \xi_{i_2}(\eta)) +$$

$$\sum_{i_1=1}^{\infty}\sum_{i_2=1}^{i_1}\sum_{i_3=1}^{i_2} b_{i_1 i_2 i_3}\Phi_3(\xi_{i_1}(\eta), \xi_{i_2}(\eta), \xi_{i_3}(\eta)) + \cdots$$

where $X(\eta)$ represents the random variable, $b_{1,2,3\ldots}$ represent deterministic coefficients, Φn represents the Askey-chaos orthogonal polynomials of order n, and ξ_i represents the set of uncorrelated random variables. Φ is chosen based on the desired distribution type for X. The different types of distributions that can be used are summarized in Table 1.

Table 1: List of Askey Chaos Polynomials Used to Represent Various Distribution Types

Distribution Type	Askey Chaos Polynomial
Gaussian	Hermite
Gamma	Laguerre
Beta	Jacobi
Poisson	Charlier
Uniform	Legendre

Most of the time random distributions can be represented by Gaussian distributions, as is the case in this project. There for hermite polynomials will be used in PCE applications. A sample of the set of hermite polynomials is shown in equation 9.

$$X(\eta) = b_0 + b_1\xi + b_2(\xi^2 - 1) + b_3(\xi^3 - 3\xi) + b_4(\xi^3 - 6\xi^2 + 3) + \cdots \quad (9)$$

Because the only undetermined portion of equation 9 are the coefficients, *b*, PCE allows for representing a complex set of arbitrary distributions using a small number of variables [12][31].

2.5 Stochastic Upscaling

The purpose of stochastic upscaling is to homogenize the input parameters to a fine scale, detailed model by creating a coarse scale model and changing the input parameters to the coarse scale model until the results are matching. The fine scale model and coarse scale model can be represented using equation 10,

$$Y_F(\omega) = f(X_F(\theta))$$
$$Y_C(\tilde{\omega}) = g(X_C(\theta)) \quad (10)$$

where Y_F and Y_C represent the outputs of the fine and coarse scale models, respectively, X_F and X_C represent the inputs to the fine and coarse scale models, respectively, and f and g represent black box models for the fine and coarse scale models, respectively. The goal of stochastic upscaling is to change X_C until Y_C matches Y_F. By reducing the complexity of

the coarse scale model, g, and reducing the number of inputs for X_C, the complexity of the model can be greatly reduced while still maintaining accuracy in the results [12][32].

The matching of results is done through an optimization process. There are several different objective functions that can be used, but the exponential difference is used in this project as the objective function. Using exponential difference allows the optimizer to more quickly find a solution, due to the steepness of the curves [32]. This optimization statement can be summarized in equation 11

$$X_C(\hat{\eta}) = \arg\max e^{-(Y_C - Y_F)^2}$$
$$\text{s.t. } |M_C^t - M_F^t| \leq \varepsilon \tag{11}$$

where M_C^t and M_F^t represent the t[th] statistical moment of the fine and coarse scale responses, respectively, and ε represents the user defined error. As shown in equation 11, in addition to the exponential difference objective function, a constraint can be optionally added to help the optimizer find an adequate solution.

Since stochastic upscaling is used for probabilistic analysis, the input variable to the coarse scale model, X_C, is often represented by a distribution. In order to efficiently model this distribution with low dimensionality, PCE is introduced. In this way, the input can be modeled using equation 8. Integrating the PCE representation of the input, X_C, and simplifying yields the representation of the optimization in equation 12.

$$U(b) = \frac{1}{n}\sum_{j=1}^{n} exp\left\{-\left(g^j\left[\sum_{k=0}^{P} b_k \Phi_k(\xi(\hat{\eta}))\right] - Y_F^j\right)^2\right\} \quad (12)$$

Using this representation, the optimization can be written with a simple number of inputs, represented by the coefficients of PCE, b_k. It is these coefficients that will be modified until the distributions of the responses for the coarse and fine scale model match each other [33].

2.6 Image Filtering

In this project, a CT scanner is used for material defect uncertainty quantification. When the CT scanner is used on a sample, the result is a stack of 2D images that can be put together for a 3D representation. These images are grayscale in nature, with values ranging from 0 to 255. Usually when scanning a specimen for defects, the image is meant to be binary, i.e. there is either material in a pixel or no material. Unfortunately, when analyzing the image stack for micro-defects, there is a lot of noise that exists that can obstruct the interpretation of the images. Due to this noise, a cutoff value cannot be accurately determined to segment the image. A filtering process must first be applied to the image stack before choosing cutoff value. The filtering process is designed to distinguish noise from actual data points being read by the CT scanner. Noise in CT scanned images is a common problem, especially at high resolutions, so many different filtering techniques have been investigated in order to accurately remove noise from the images. Davidoiu et al. investigated different filtering techniques by manufacturing a "clean" CT scanned image and introducing artificial noise. Various filtering methods were then used to attempt to remove the noise from the system to get back to the original image with varying degrees of success [34]. Although the authors recommended the isotropic

total variation (ITV) method for filtering CT scan images, in practice it was found that Gaussian filtering produced good results as well. This decision was made based on comparing the Gaussian filtered images to existing images of CT scanned parts [35]. Gaussian filtering works by comparing a pixel to the pixel values in a block around it. The size of this block is arbitrary and depends on the user. After selecting an appropriate block size, equation 13 is used to calculate a new value for the pixel,

$$h(x,y,z) = \frac{1}{2\pi\sigma} e^{\left(\frac{-[x^2+y^2+z^2]}{2\sigma^2}\right)} \tag{13}$$

where h is the calculated value, and σ represents the standard deviation of the block. This process can be repeated multiple times to achieve the desired effect [36].

The ITV method works based on the total variation of the system, represented as the Euclidean norm of the gradient. The ITV filtering process is proposed as a minimization problem shown in equation 14,

$$f(u) = \text{argmin}\left\{\frac{1}{2}\|u-z\|_2^2 + \alpha|u|_{TV} : u \in \mathbb{R}^d\right\} \tag{14}$$

where u is the desired solution, z is the data value to be denoised, α is the regularization parameter, and $|u|_{TV}$ is equal to $\|\nabla u\|_2$. The processed image will have varying degrees of denoising based on the chosen value of α.

2.7 Bayesian Information Criterion

Bayesian information criterion (BIC) is a method for selecting an appropriate distribution for a set of data. There are several different types of distributions that can be

used to represent data, including Gaussian, extreme value, gamma, etc. BIC uses information loss to choose the most appropriate model for the distribution, as shown in equation 15,

$$IL(\omega_j|X) = -2\ln p(X|\hat{\theta}_j, \omega_j) + k_{p,j}\ln(n) \quad (15)$$

where IL represents the information loss, ω_j represents the jth distribution model, X represents the input data, θj represents the input parameter for the jth model, kp,j represents the number of estimated parameters in the jth model, and n represents the number of data points. The model with the most amount of information loss is chosen as the best fit [37][38].

The BIC method for choosing distribution is advantageous over many other methods because it takes into account the number of samples, *n*. Taking in the number of samples as in input allows the method to work for distributions of varying sizes with appropriate bias. In addition BIC often has a faster convergence rate over other methods [39][40].

2.8 Validation Methods

Validation is the process of comparing the results of a model to experimental results to ensure that the model is an accurate representation of reality. There have been many different validation methods suggested for probabilistic validation, all with varying advantageous and disadvantages.

2.8.1 Single Site and Multiple Site Validation

When performing validation of probabilistic responses, it is important to distinguish between single site and multiple site validation. For an example, take a simple beam problem as shown in Figure 2-3, where a beam is fixed at one end and has a point force on the other end.

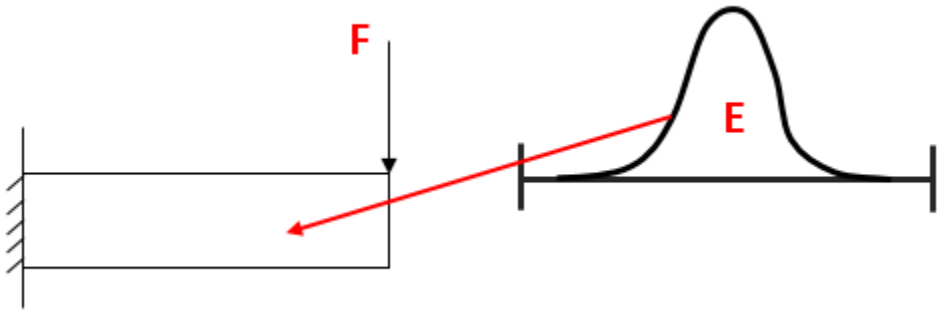


Figure 2-3: Beam with Varying Elastic Modulus

In this problem the elastic modulus (E) of the beam is considered to have aleatory uncertainty, thus the displacement of the end of the beam will vary depending upon the elastic modulus. Single site validation is when only one input is considered, i.e. the point force is equal to 5 N. Multiple site validation would be applicable if one wanted to validate the displacement of the beam at many different force values. This is markedly different than validation with multiple responses, which might attempt to validate both the displacement and maximum stress in the beam.

In order to allow for validation at multiple sites, a technique called u-pooling is employed. U-pooling is the process of transforming an arbitrary CDF into a uniform CDF. An example of this process can be seen in Figure 2-4.

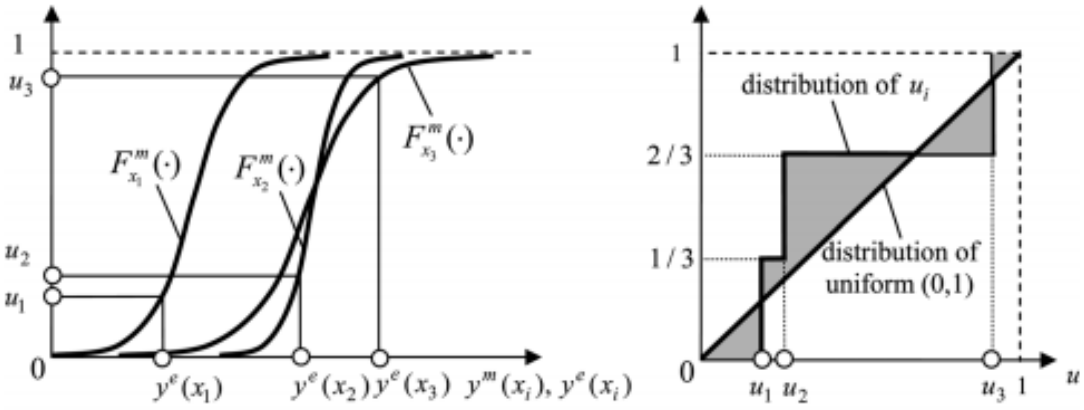


Figure 2-4: Example of U-Pooling

Once the function to transform the model results to a uniform CDF is obtained, that same function is used to transform the experimental results. This process can be completed for each validation site to transform each set into a range from 0 to 1. Afterwards, and type of validation method for aleatory uncertainty can be applied [25].

2.8.2 Validation for Aleatory Uncertainty

Many different validation methods have been proposed for aleatory uncertainty. In this section, some important ones will be presented along with their advantages and disadvantages [25][41][42].

2.8.2.1 Area Metric

The area metric is a method developed to attempt to compare two CDF's in their entirety. It works by graphing the two distributions, and adding up the area in between the two distributions, as shown in Figure 2-5.

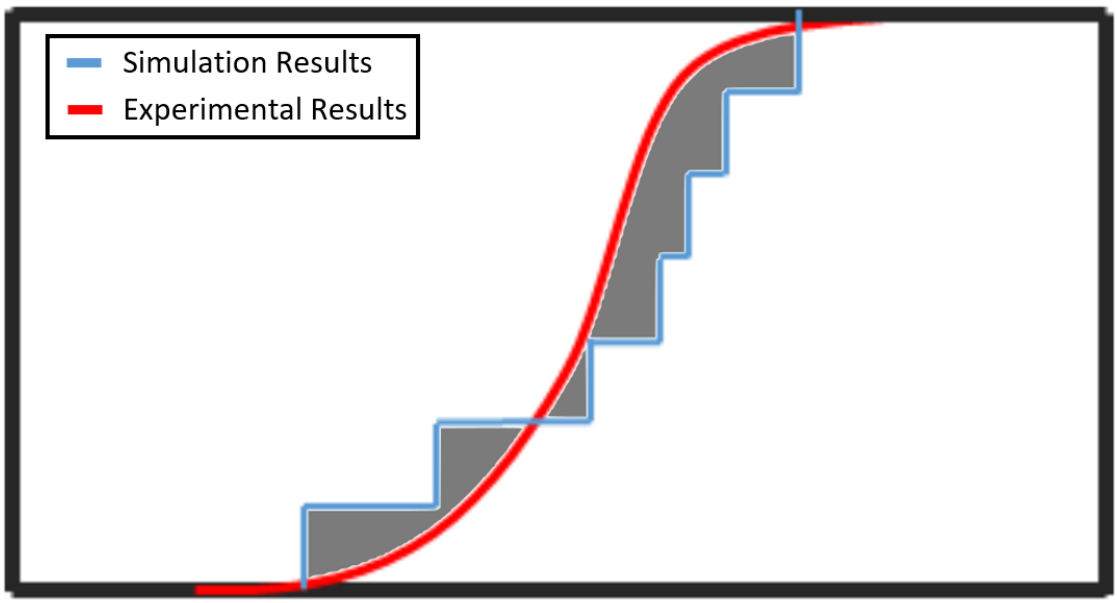


Figure 2-5: Area Metric Example

Once the value of the area difference is calculated, it is then compared to another pre-determine value. If the area is less than the pre-determined value, then the model is accepted.

The largest disadvantage to the area metric is that the pre-determined value to which the metric is compared is arbitrary. Since CDF's can exist at any values, there is no significance level attached to a comparison. Therefore it is impossible to say that the model is accepted within a certain degree of certainty. However, if the goal is to compare two different models by validation with experimental results, the area metric can effectively show which model is a better fit for reflecting reality. In addition, the area metric does take into account the entire distribution when calculated, making it effective when comparing two different models to each other [25].

2.8.2.2 Frequentist Metric

The frequentist metric compares the mean of two distributions. The standard deviation is used to create a confidence interval based on the experimental data as shown in equation 16

$$\hat{e} - t_{\frac{\alpha}{2}}(N-1)\frac{s}{\sqrt{N}}, \hat{e} + t_{\frac{\alpha}{2}}(N-1)\frac{s}{\sqrt{N}} \tag{16}$$

where $\hat{e}$ represents the error in the model, $t_{\alpha/2}(N-1)$ represents a value obtained from the t-distribution based on the significance value α/2 and the number of samples, *N*, and *s* represents the standard deviation. The mean value of the model is then compared to the confidence interval, and if it lies inside of it then the model is accepted.

The frequentist metric is advantageous since it relies on the well-developed t-distribution, as well as effectively providing a metric accounting for both a significance value and the number of samples. The frequentist metric suffers from the fact that it only compares the mean of the two distributions, therefore not taking into account the variation of the model or a comparison of the entirety the two distributions [25][41][43].

2.8.2.3 Kolmogorov-Smirnov Test

The Kolmogorov-Smirnov test (K-S test) is a method developed to test if two distribution come from same parent distribution. This is done by calculating the maximum distance between the two CDF's and then comparing that value to a developed statistical metric associated with a significance value. This statistical metric is generally calculated from a table given the number of samples and desired significance level.

The largest disadvantage to the K-S test is that it only takes into account the maximum distance between the distributions. Although this is slightly affected by the entire distribution, two sample models can have the same maximum distance when compared to experimental results, while one might be a much closer match across the entire distribution. The advantage to using K-S test is that the calculated metric can be compared to a value that has statistical significance, reflecting on the confidence of the result. In addition, it takes into account the number of samples, making it effective for distributions of varying size [25][44][45]. In this project, the K-S test will be used for validation with aleatory uncertainty due to its ability to moderately account for the whole distribution while still allowing for a metric to be calculated based on a significance value.

2.8.3 Validation for Aleatory and Epistemic Uncertainty.

There are relatively few methods available for validations under both epistemic and aleatory uncertainty. The difficulty arises because in cases of aleatory and epistemic uncertainty, the model response is represented by a p-box, i.e. 2 CDF's, while the response of the experiments is still represented as a single CDF. There have been a few methods created to try to overcome this.

2.8.3.1 Mean Curve Method

The mean curve method attempts to overcome the difficulty by turning the p-box representation into a single curve. This is done by taking the two CDF's and calculating the mean x-value at each point, as shown in Figure 2-6a. After obtaining the mean curve of the model, the mean curve and the experimental CDF can then be compared using any available validation metric for aleatory uncertainty.

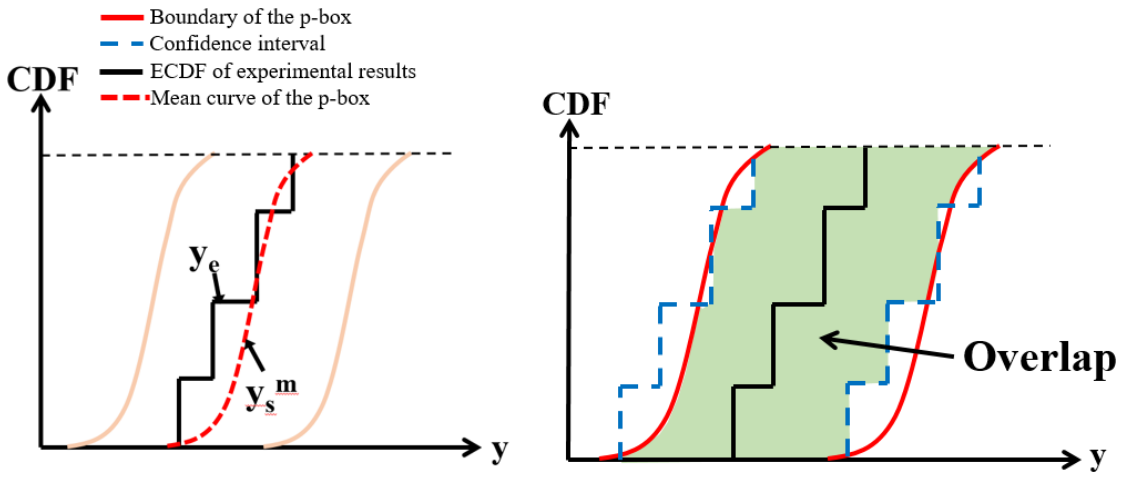


Figure 2-6: Existing P-Box Validation Methods - a)Mean Curve Method b) Confidence Interval Method

The largest drawback to the mean curve method is that it is possible for the experimental response to be located close to the edge of the p-box while still being inside of the p-box. In this case, the p-box is an accurate representation of the results, since the experimental response lies inside of it, but the mean curve will be far different from the experimental response, most likely resulting in rejection [46].

2.8.3.2 Confidence Interval Method

The confidence interval method has a different approach to the problem. By using the standard deviation of the experimental response, a confidence interval around the experimental results can be created. Then the overlap ratio between the confidence interval and the p-box can be taken. If the overlap ratio is higher than a pre-determined value, then the model is accepted. This process can be seen in Figure 2-6b.

The largest drawback to the confidence interval method is that even though there is a significance level attached to the confidence interval, the value to which the overlap ratio is compared to is arbitrary. In addition, if the width of the p-box is to large, it is possible that the model will still be rejected even if the experimental response lies inside of the p-box [46][47].

2.8.3.3 Resimulation Method

The third method is the resimulation method. This method takes advantage of the fact that a p-box is technically made up of several different CDF's, as shown in Figure 2-7a. Normally these are ignored due to epistemic uncertainty, but in this method, the experimental response is compared to each of the individual CDF's using the area metric. Since each individual CDF is associated with a certain epistemic input, a data set can be created that relates the area metric value as a function of the epistemic input value. A kriging model is then created to approximate this function, and an optimization algorithm can be used to find the optimum epistemic input. This process can be seen in Figure 2-7b. Finally, the model is then re-simulated using the optimum epistemic input and the resulting response can then be validated against the experimental response using any validation metric for aleatory uncertainty. This final validation process can be seen in Figure 2-7c.

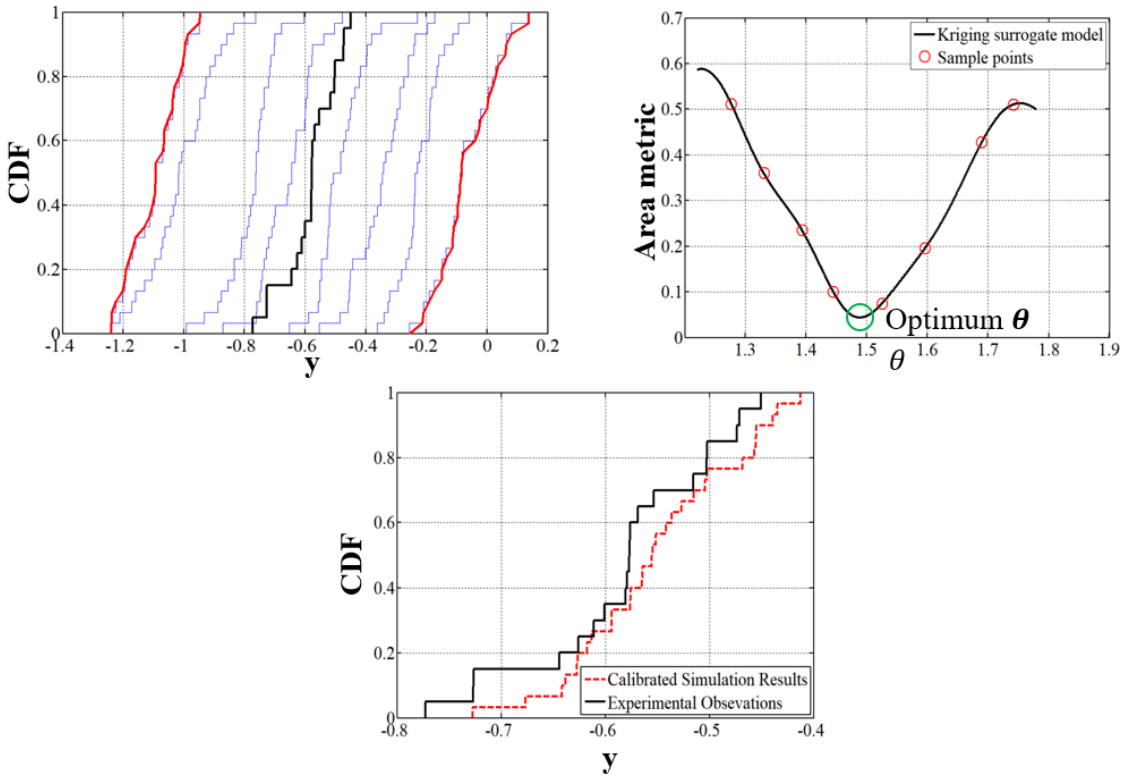


Figure 2-7: Proposed P-Box Validation Method - a) P-Box Comprised of Multiple CDF's b)Kriging Model to Find Optimum Input Epistemic Variable c) Final Validation Process

The largest drawback to this validation method is that obtaining enough CDF responses from the model to create an appropriate kriging model can be extremely computationally expensive. In addition, the method requires an optimization step followed by another set of simulations which can make the process less computationally efficient. The largest benefit to this method is that as long as there is sufficient model data, the resulting metric should accept a correct model. In addition, it makes use of the area metric's strong ability to compare entire distributions while mitigating its disadvantage of having no significance level.

CHAPTER 3. GEOMETRIC UQ FRAMEWORK

The purpose of the geometric UQ framework is to quantify the uncertainty of geometric uncertainties in a manufactured part, and propagate those uncertainties into simulation. The framework is split into 4 steps. Step 1 is to create a manufactured specimen and analyze the error between the manufactured part and CAD part. Step 2 is to create a fine scale finite element model that incorporates the uncertainty data obtained from step 1 and perform fine scale simulations. Step 3 is to create a coarse scale model with greatly reduced complexity. Finally, step 4 is to introduce an input variable to the coarse scale model and use upscaling to match the response of the coarse scale model to the response of the fine scale model. The organization of this chapter will reflect these four steps. A flow chart of this process can be seen in Figure 3-1. For this framework, geometric uncertainty will be considered as aleatory uncertainty, therefore the resulting responses of the coarse and fine scale models will be CDF's.

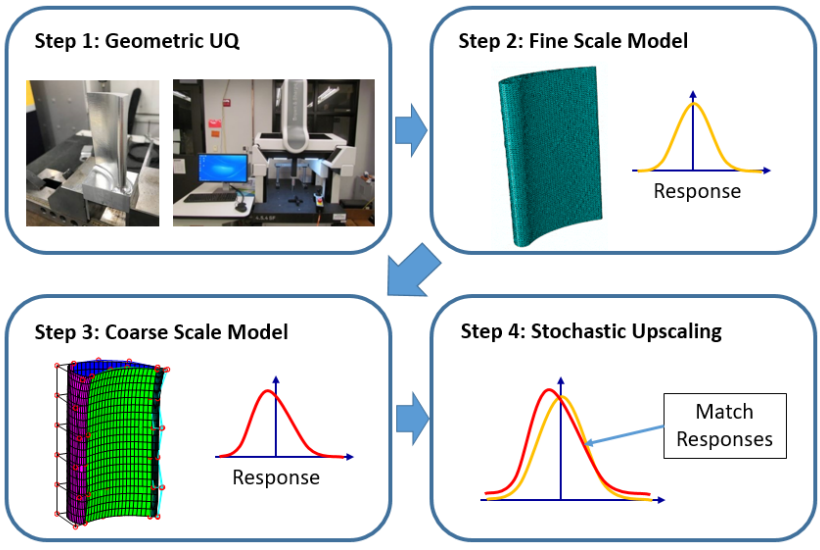


Figure 3-1: Flow Chart of Geometric Uncertainty Quantification Framework

3.1 Geometric Uncertainty Quantification

The first step is manufacture a part and quantify the error between the manufactured part and the CAD model. Turbine blades are typically either casted or machined. It is important to use the same process that is used to manufacture the actual turbine blade. After manufacturing the part, it should be scanned by machinery that is capable of detecting the small differences between the CAD model and the manufactured part in 3D space. This can be done by a coordinate measuring machine (CMM) or equivalent machine. When a CMM machine measures a part, it is first calibrated with the CAD model to zero the machine on the desired origin. Then the surface of the part is discretized into a field of points. For each point, 3 types of data are recorded: the point on the CAD model, the normal vector associated with that point, and the point on the manufactured part that lies on the normal vector. By taking the distance between the point on the CAD model and the point on the manufactured part, the geometric error can be found. If the point on the manufactured part exists on the inside of the CAD model, then the error is taken to be negative, and if it exists on the outside of the CAD model, then the error is taken to be positive.

After obtaining the difference between the CAD model and the manufactured part, the next step is to use semivariogram analysis to construct a covariance matrix for the field of points. Using equation 2 from section 2.2, a semivariogram can be constructed. In the equation, z represents the variable that is being analyzed, which in this case is the error between the CAD part and manufactured part. The lag distance, h, is calculated by finding the distance between the two points via the surface of the part. In some cases, some sections

of a part may be considered uncorrelated with each other. In this case, these sections can be separated and a separate semivariogram analysis can be performed for each section.

Finally, after all the semivariograms for the part have been constructed, the least squares method should be used to fit a model to the semivariogram data. This way the semivariogram value, and therefore the covariance, can be calculated for an arbitrary lag distance. This relationship will be used for constructing the fine scale model.

3.2 Fine Scale Model Simulations

The next step is to construct the fine scale finite element model, propagate the uncertainty into the finite element model via K-L expansion, and perform stochastic simulations to obtain a distribution for the required response.

3.2.1 Fine Scale Model Construction

The fine scale finite element model can be constructed in two different ways. The simple method is to simply use the CAD model from which the part was manufactured. This can be done by importing the CAD model into any common FEA software and meshing using the software's meshing capabilities. The purpose of the fine scale model is to model small details in the geometry of the part, so the mesh should be suitably fine. Alternatively, if the data from the CMM machine is suitably detailed, the surface points measured for the manufactured part can be used to interpolate a surface model. In this way the fine scale model will more accurately represent reality. After constructing the mesh for the fine scale model, the desired boundary conditions should be applied.

3.2.2 Uncertainty Propagation

The uncertainty quantified from section 3.1 can be propagated into the fine scale finite element model using K-L expansion. K-L expansion works based on an eigenvalue decomposition of the covariance matrix, which can be obtained from the semivariogram functions obtained in section 3.1. This can be done by considering each individual surface point on the finite element model as its own variable. This can be shown in matrix form using equation 17

$$p = [p_1, p_2, p_3 \dots p_n] \tag{17}$$

where p_i represents the i^{th} surface point. In this way, for a finite element model with n surface points, the constructed covariance matrix (Λ) will be an $n \times n$ matrix calculated using equation 18,

$$\Lambda_{i,j} = Sill - \hat{\lambda}(h_{i,j}) \tag{18}$$

where $\Lambda_{i,j}$ represents the covariance of the geometric error for points p_i and p_j and $\hat{\lambda}$ is the semivariogram value for the lag distance $h_{i,j}$, which represents the distance between points p_i and p_j calculated via the surface of the part. The sill is simply the highest value in the corresponding semivariogram. After obtaining the covariance matrix, the standard Gaussian distribution can be used to generate a set of uncorrelated random variables. Using K-L expansion with the covariance matrix, a correlated set of random errors can be calculated for each surface point, p_i.

3.2.3 Stochastic Simulations

The final step is to perform stochastic simulations with finite element model. For a single simulation, K-L expansion should be used to create a new set of correlated random errors. These errors are then applied to the surface coordinates by moving each point by its prescribed error in the direction normal to the surface. It is important to check if modifying the coordinates produces badly shaped elements. If so, the mesh might have to be slightly modified every simulation. After modifying all the surface coordinates, the simulation should be performed and the desired response recorded. This process can be repeated an arbitrary number of times until a CDF of the response can be constructed.

3.3 Coarse Scale Model Construction

The purpose of creating the coarse scale model is to create a finite element model that has a similar geometry to the fine scale simulations, but with greatly reduced complexity. In most cases, this would result in a less accurate model. However, after performing the stochastic upscaling process, the response of the coarse scale model should reflect the results of the more accurate fine scale model. This can be useful for many applications including modelling subsystems and performing reliability analysis. For example, in the turbine blade example, the coarse model of the turbine blade could be substituted into a larger turbine engine finite element model, and then stochastic simulations of the entire engine could be performed at greatly reduced computational cost. In addition, many times it is desirable to do reliability analysis, where the chance of failure is analyzed through stochastic simulation. When the desired reliability is extremely high, this can take hundreds of thousands of simulations, while the upscaling process can work with much less. Thus the reliability analysis can then be performed with the coarse scale model with greatly reduced computational cost. The coarse scale model construction can

be split into two different steps. The first step is to approximate the surface of the manufactured part using Bezier curves and surfaces or other similar surface modelling techniques. The second step is then to turn the surface model into a finite element model and apply simulation parameters.

3.3.1 Surface Approximation

The surface of a part can be approximated in many ways. The first obvious solution is to use the same fine scale finite element model, but reduce the density of the mesh. This method is similar to other methods of discretizing the surface into a grid of points and using that to create a new mesh [48]. These processes are easy to do, but reduce the amount of options available for introducing inputs in the stochastic upscaling step. Due to the large number of points (even with a coarse scale model), it is impossible to use the surface coordinates as an uncertainty input. This problem can be slightly mitigated in a turbine blade example by using discrete points to model several 2D turbine blade profiles along the height of the blade, as shown in Figure 3-2a [49][50].

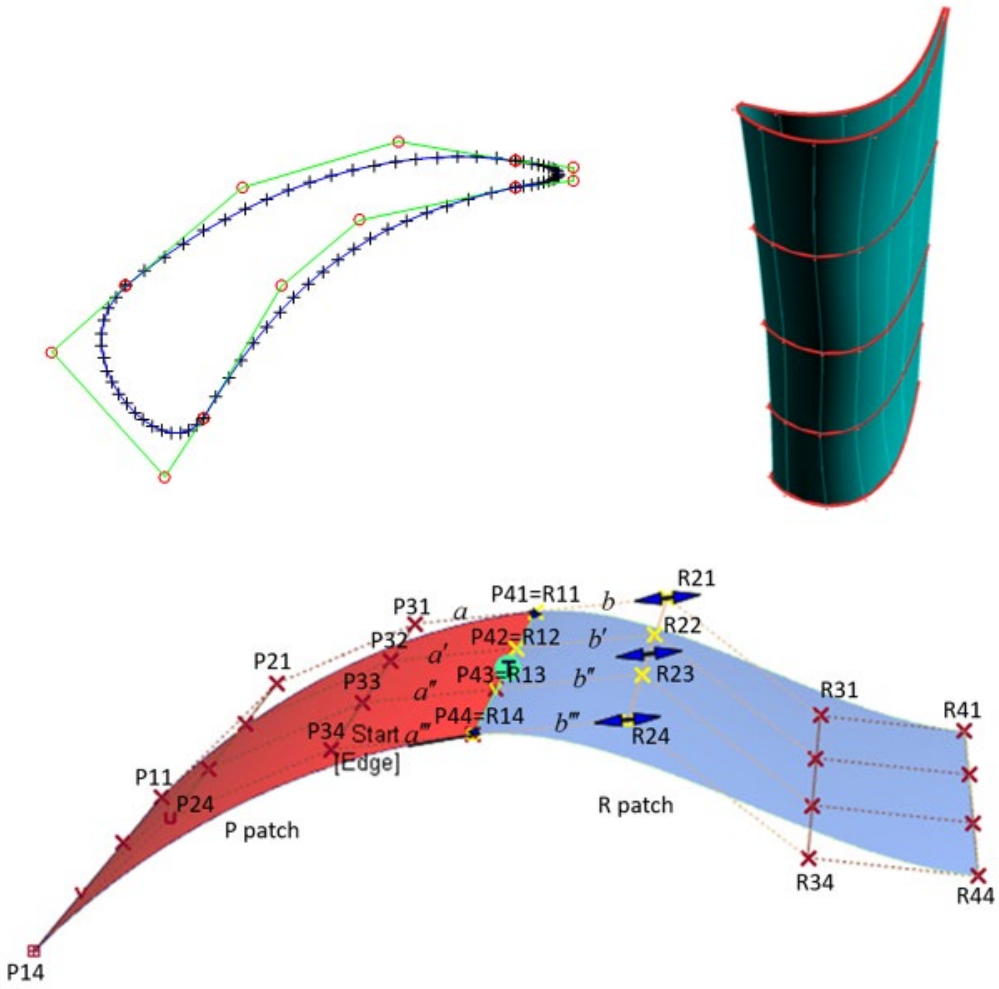


Figure 3-2: Bezier Surface Creation - a)Single Blade Profile b) Blade Surface Representation c) G1 Continuity

The profiles can then be lofted to create a full surface representation of the turbine blade, as shown in Figure 3-2b. This helps reduce the complexity greatly, but the complexity can be further reduced by using parametric modelling techniques to approximate the 2D turbine blade profiles. There are several parametric modelling techniques available for this type of problem, such as Bezier curves, Hermite curves, B-splines, and NURBS [51][52]. These allow for approximation of the 2D turbine blade profiles using a very few sets of control

points. Modelling the part via parametric curves then allows those control points to be used as possible inputs during upscaling. For this framework, Bezier curves were chosen to model the turbine blade do to its simplicity. In addition, Bezier curves are very easy to extend to Bezier surfaces, which can be used in place of the lofting process or for modelling any arbitrarily shaped part [53][54].

A Bezier curve of n^{th} degree can be shown in equation 19,

$$\boldsymbol{P}(u) = \sum_{i=0}^{n} B_{i,n}(u) \cdot \boldsymbol{P}_i \tag{19}$$

where $\boldsymbol{P}_i$ represent the control points, and $B_{i,n}$ are the Bernstein polynomials, which are calculated via equation 20,

$$B_{i,n}(u) = \binom{n}{i} \cdot u^i (1-u)^{n-i} \tag{20}$$

where $\binom{n}{i}$ is the combination of i from k, represented by $\frac{n!}{i!(n-i)!}$. Bezier curves are also easily extended into 3D Bezier surfaces by using equation 21,

$$\boldsymbol{P}(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} B_{i,n}(u) \cdot B_{j,m}(v) \cdot \boldsymbol{P}_{i,j} \tag{21}$$

where $\boldsymbol{P}_{i,j}$ are the control points of the control polyhedron of the Bezier surface, and u and v are values in the range of 0 to 1. When using several Bezier curves and surfaces to model a continuous surface, it is often important to ensure G^1 continuity between the surface, i.e. the derivative of the surface does not change suddenly when going from one Bezier surface

patch to another. This can be done by ensuring that control points for a Bezier surface that lie next to a boundary with another surface are collinear with the nearby control point on that surface and the shared control point on the boundary [55]. This process can be seen in Figure 3-2c.

3.3.2 Finite Element Model Creation

The next step is to construct a finite element model based on the surface approximation obtained from the Bezier surfaces. Because Bezier surfaces are parametrically defined, it is fairly simple to generate a grid of values for *u* and *v* and plug it into the surface equations to generate a grid of points that lie on the surface of the part. Then, each Bezier surface can be triangularly meshed using 2D Delaunay Triangulation [56]. Delaunay Triangulation is an algorithm that can generate unique set of triangles whose vertices are points in 2D space. It is defined as the set of connections that form triangles such that no point of a triangle lies inside the circumcenter of another triangle. An example of this can be seen in Figure 3-3a.

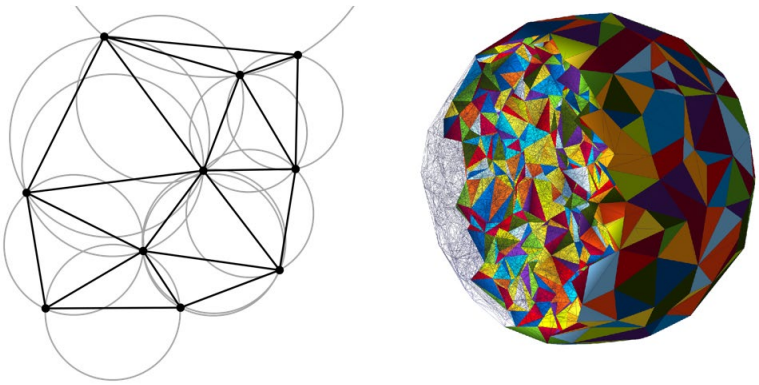


Figure 3-3: Delaunay Triangulation - a) 2D Delaunay Triangulation b) 3D Delaunay Triangulation

Since this process is strictly used for 2D sets of points, and Bezier surfaces are usually 3D in nature, the process must be modified slightly. The Bezier surface can be flattened into 2 dimensions such that the distance between two points on the Bezier curve in 3D space calculated strictly via the surface is the same as the distance between the points on a 2D plane. Then 2D Delaunay triangulation can be used to generate the set of nodes for the triangular mesh while using the 3D points as the point set. Completely defining the surface mesh and ensuring consistency in the definition of the triangular elements allows there to be a clear distinction between the space that lies inside of the part and outside the part. Then points can be added to the inside of the part and finally a tetrahedral mesh can be generated using 3D Delaunay Triangulation. This method works similarly except it is defined as the set of tetrahedra connecting points such that the no point of any tetrahedra lies inside the circumsphere of another tetrahedra. This method is preferred way of generating a 3D mesh due to its ability to often form sets of tetrahedra that are approximately equilateral [57]. This process can be seen in Figure 3-3b.

3.4 Stochastic Upscaling

The final step is to introduce an input (or multiple inputs) to the coarse scale model and use stochastic upscaling to match the response to the fine scale model. An overview of the stochastic upscaling process for geometric UQ framework can be seen in Figure 3-4.

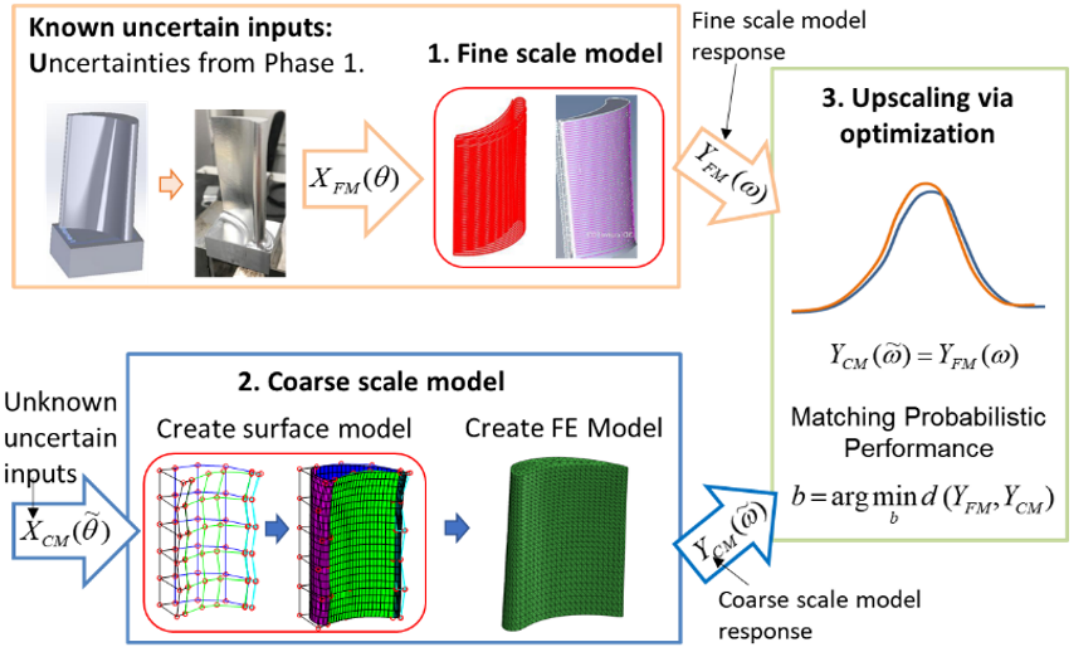


Figure 3-4: Stochastic Upscaling Process

3.4.1 Optimization

Once the inputs are chosen, the desired order of PCE should be chosen. Lower order PCE's allow for quicker and easier optimization, but often result in less accurate results. In general, a PCE of order 2 or 3 maintains a balance between accurate result and ease of optimization. The inputs for the optimizer are then given as the coefficients of PCE. The objective function should take the coefficients for PCE, generate a distribution for the input variables, perform stochastic simulations of the coarse scale model, and then compute a value from the objective function as shown in equation 11. For each set of coarse scale simulations, the number of simulations should be the same as the amount of simulations performed with the fine scale finite element model.

3.4.2 Surrogate Modelling

Computing a large number of simulations for each iteration of the objective function can be extremely time consuming. This problem is exacerbated by the fact that form iteration to iteration, it is very likely that the optimizer will be performing near identical simulations several times. In order to get past this, it is possible to use a surrogate model to predict the response of the coarse scale model based on the inputs. This way the objective function can be computed much more quickly, greatly reducing the computational cost of stochastic upscaling.

CHAPTER 4. MATERIAL UQ FRAMEWORK

The purpose of the material UQ framework is to quantify the material defect uncertainty associated with manufacturing, and integrate it into simulation. More specifically, it will begin by collecting data on porosities from a CT scanner and integrating it into a simple fine scale beam model. Stochastic upscaling will then be performed in order to obtain a homogenized elastic modulus. Since material defect uncertainty exists on a micro scale level while the geometric uncertainty exists on macro scale level, material defect uncertainty is impossible to integrate directly into the fine scale model from chapter 3. By performing upscaling to extract a homogenized elastic modulus, the material defect uncertainty can be propagated into a macro scale level to extract a probabilistic response. The framework can be broken down into 6 steps. Step 1 is scan the material with a CT scanner and collect data on the porosities inside of the material. Step 2 is to use the collected data to create fine scale models. In this case, the fine scale model will be a simple rectangular beam model. Step 3 is to create a coarse scale model and perform upscaling to extract a homogenized elastic modulus. Step 4 is to use the obtained elastic modulus values to modify the elastic modulus values of the fine scale simulation obtained in chapter 3 and obtain a probabilistic response. Step 5 is to create a new coarse scale model of the desired part and introduce inputs. Finally step 6 is to perform upscaling so that the response of the coarse scale model matches the result of the fine scale model. For this framework, material uncertainty will contain epistemic uncertainty, therefore the inputs and responses will be modeled with intervals and p-boxes.

4.1 Analyzing CT Scan Data

The goal of this step is to scan a sample of material using a CT scanner and extract information from the images about porosity information. An overview of this process can be seen in Figure 4-1.

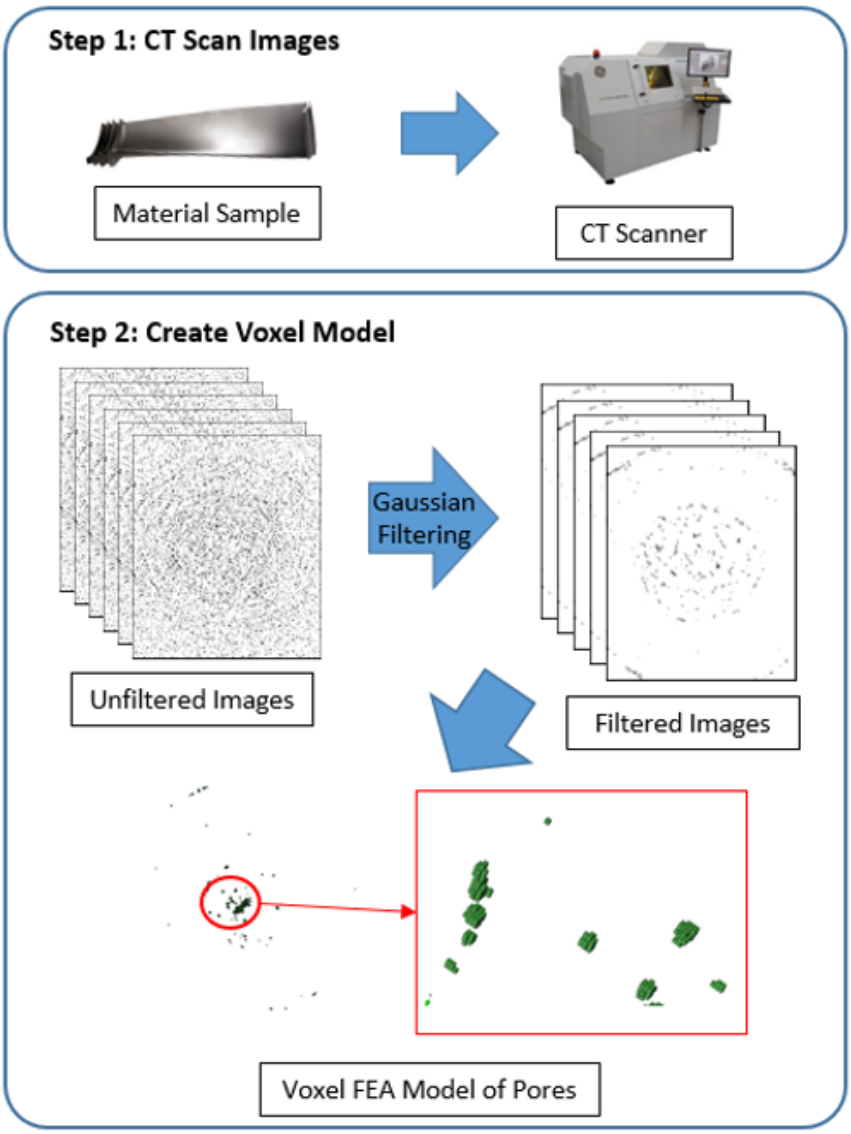


Figure 4-1: Porosity Information Extraction Process

4.1.1 Interpreting CT Scan Data

When a CT scanner scans a sample of material, the data is constructed as many 2D images that can be stacked on one another to form a 3D image representing the scanned material sample. In these images, white is used in a pixel where material is present, and black is used in a pixel where material is not present, i.e. a pore exists. However, the results obtained directly from the CT scanner are not binary images. Rather, they are greyscale images where each pixel contains a value between 0 and 255, where 0 is completely black and 255 is completely white. Thus, a cutoff value must be determined in order to separate light pixels from dark pixels. For example, if a cutoff value of 122 is chosen, any value below that will be determined to be absence of material, and any value above that will be determined to be to contain material. Before this cutoff value can be chosen though, the images must first be filtered. When using modern CT scanners, there is often an issue with noise plaguing the image, as shown in Figure 4-1. This noise is often an increased problem when higher resolustions are used. To remove this noise, many filtering options are available, but it has been determined that Gaussian filtering and ITV filtering produce the best results. Filtering is a very important step, as shown by difference between the filtered and unfiltered images in Figure 4-1. Once the images have been filtered, a suitable cutoff value can then be chosen to segment the images into a binary image.

4.1.2 Extracting Porosity Information

When quantifying the porosity in the CT scan image of the sample material, it is often useful to first construct a voxel model. This can be approached in two different ways: constructing a voxel model where each voxel represents material, or constructing a voxel

model where each voxel represents a pore. The main advantages of the first method is that it is more intuitive to view a voxel model where voxels represent physical material (since this is how voxel models are typically used), and that information on all of the boundaries of the material sample are kept in the voxel image. However, in practical use, it was found that constructing, viewing, and performing analysis on such a voxel model proved extremely computationally expensive. Thus the images were instead converted to a voxel model where the pores were represented by voxels. Since there are typically significantly less pores than material, this greatly reduces the computational complexity. To conserve the information on the boundaries of the sample, it was sectioned off such that the shape of the sample was a cube. Thus the boundary information could be easily recorded separately and used later for analysis. An example of such a voxel model can be seen in Figure 4-1. The size of these voxels is determined based on the resolution of the CT scanner.

The next step is to collect information on the porosity of the material. In this framework, information on the overall density of poses as well as the size of the pores is collected. For a single sample, the density is taken to be the fraction of the space that is occupied by pores, i.e. a value between 0 and 1, where 0 means the sample contains no pores. The voxels that do exist in the voxel model are then grouped into pores. This is done specifically by considering that any voxels that share more than 1 coordinate are considered to be part of the same pore. Once the pores are grouped, the distribution of the size of pores can be constructed using Bayesian Information Criterion.

4.2 Constructing Fine Scale Models

The next step is to construct a fine scale model to incorporate the porosity information gained from section 4.1. The created fine scale model for this framework is a simple rectangular beam model. The beam model is constructed out of voxels that are the same size as the voxel model obtained in section 4.1. The size of this rectangular beam can be arbitrary, it just must be big enough that it can fit the pores that were contained in the sampled material. Next, pore sizes are sampled from the distribution obtained using BIC, and are inserted into the beam model via deleting elements. The positions of the pores are determined randomly using a uniform distribution. The pores are continuously added until the density of the beam model matches that obtained from the sample of material. An example of this can be seen in Figure 4-2. Next the simulations parameters are added. For this simulation, the beam is fixed at one end, and a strain is applied to the other end. Then the response of the simulations is the reaction force obtained at the fixed end.

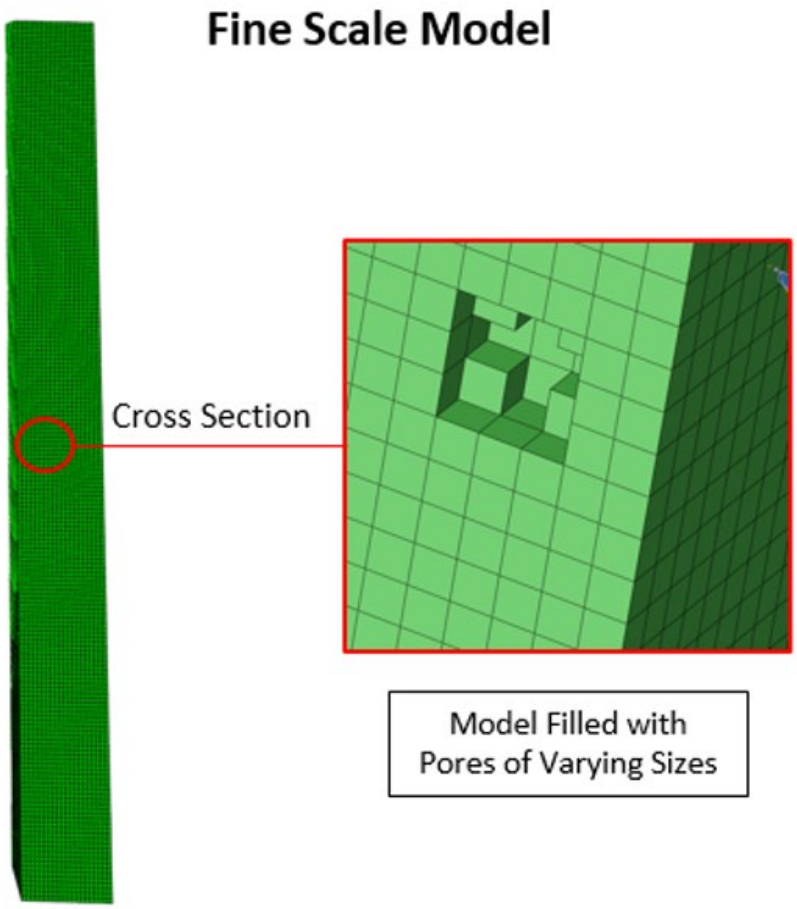


Figure 4-2: Fine Scale Model Porosity Propagation

This simulation is repeated many times, randomly inserting pores and recording the reaction force, until a distribution of reaction force is obtained. This step is repeated separately for each sample of material. For example, if *n* samples of material are scanned, then *n* different sets of pore data (density and pore size) will be recorded, *n* sets of fine scale model simulations will be conducted, and *n* distributions of reaction force will be obtained.

4.3 Stochastic Upscaling for Elastic Modulus

The next step is to create a coarse scale model, introduce an input variable, and use stochastic upscaling to find a homogenized elastic modulus. This will allow the material defect uncertainty to be propagated into a macro scale finite element model, even though the defects exist on a micro scale level.

4.3.1 Coarse Scale Model Creation

The coarse scale model for the beam problem is taken to be as simple as possible. It is a beam of the same dimensions but only consists of a single element, as shown in Figure 4-3.

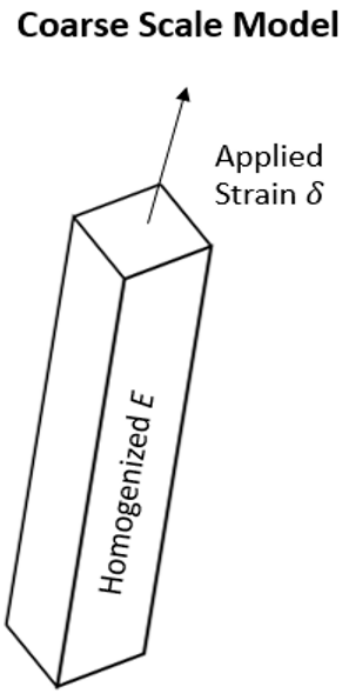


Figure 4-3: Material UQ Coarse Scale Model

The same simulation parameters are applied, so one end of the beam is fixed and the other has an applied strain. Thus, the behaviour of the coarse model is described by equation 22, and rearranging such that reaction force is the response yields equation 23,

$$\sigma = E\delta \tag{22}$$

$$F = AE\delta \tag{23}$$

where σ is the stress in the element, E is the elastic modulus, δ is the applied strain, F is the reaction force, and A is the cross sectional area of the beam [58]. Since the reaction force is the response and everything else is deterministic, the elastic modulus is the best candidate for an input variable.

4.3.2 Stochastic Upscaling Procedure

Since the elastic modulus is introduced as the input to the coarse scale model, the goal of the stochastic upscaling procedure is to find a distribution of elastic moduli so that the reaction force of the coarse scale model matches that found from the fine scale model. Similar to before, the distribution of elastic moduli are represented by PCE, and it is these coefficients of PCE that are determined in the upscaling process. A flow chart of this process can be seen in Figure 4-4.

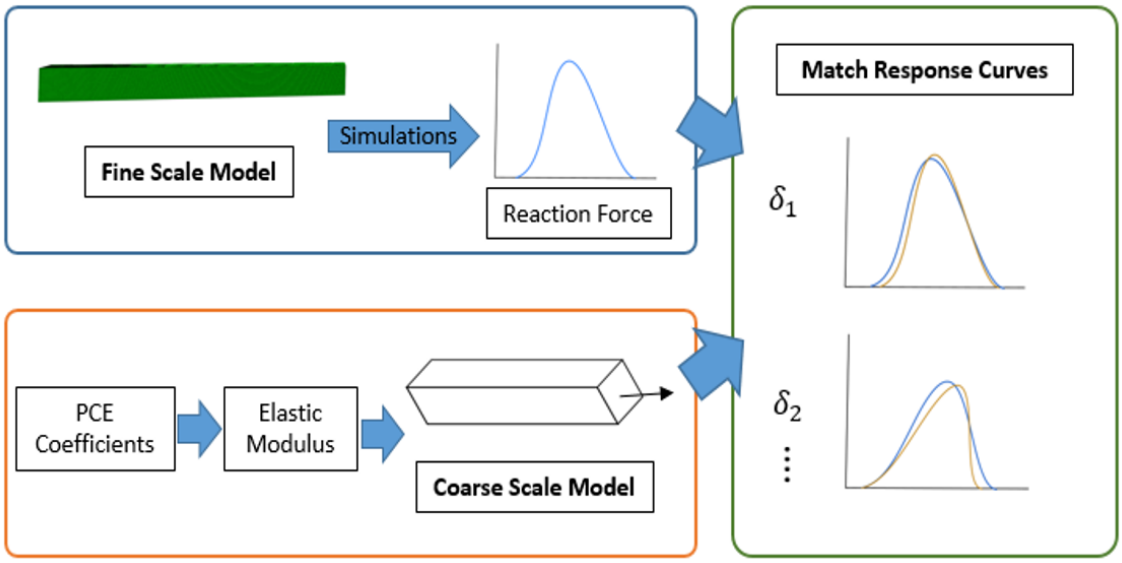


Figure 4-4: Material UQ Stochastic Upscaling Process

As mentioned before, for *n* samples, *n* different sets of fine scale simulations are conducted to obtain *n* distribution of reaction force. Thus, when upscaling is performed, *n* different distributions of elastic moduli are obtained. Often times it is difficult to obtain many samples of CT scan data, as it is an expensive and time consuming process. Thus, although there is enough information on pore size to consider it as aleatory, often times the information obtained on density must be considered as epistemic. Therefore, when the number (*n*) of elastic moduli distributions are obtained, they form a p-box representation, as shown in Figure 4-5. This p-box representation can then be used in the fine scale model obtained in Chapter 3.

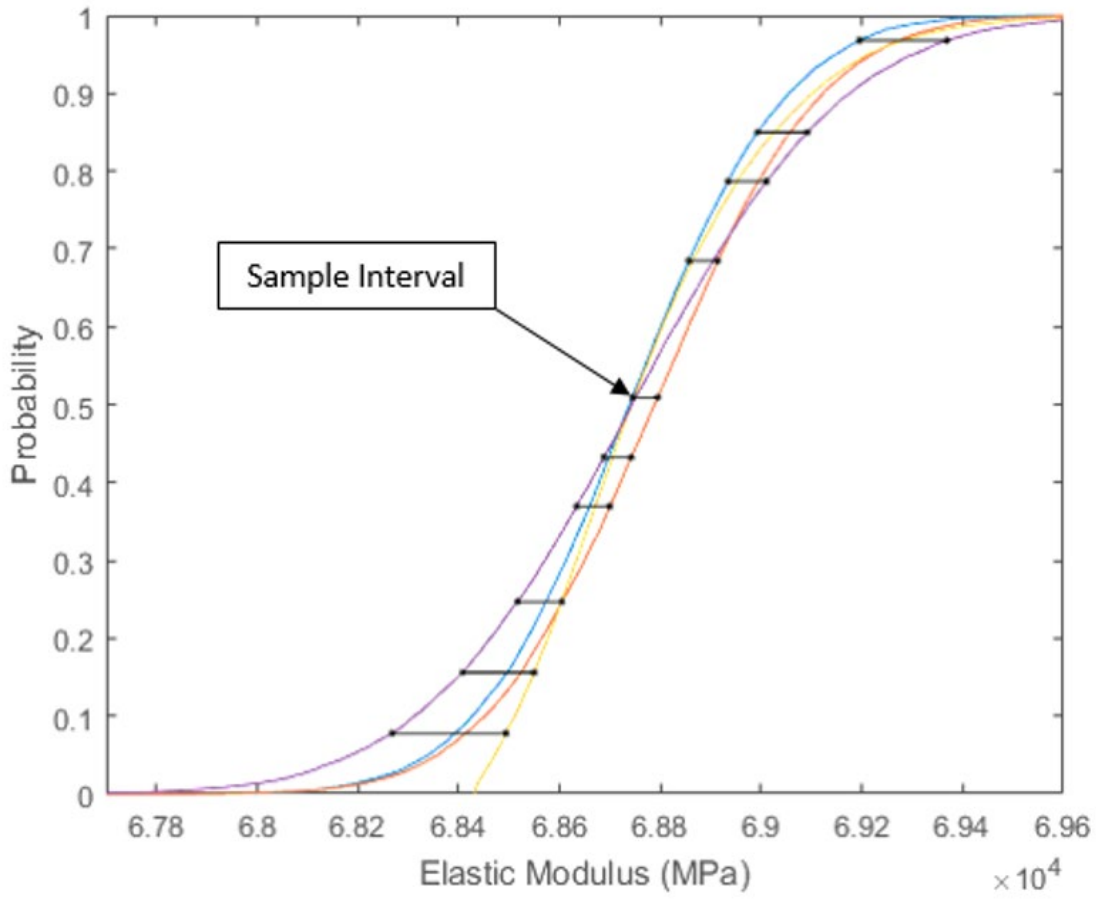


Figure 4-5: Elastic Modulus P-Box Example

4.4 Elastic Modulus Uncertainty Propagation

The next step is to propagate the uncertainty information obtained on the elastic modulus into the fine scale model of the desired part. This is done by first sampling elastic modulus values from the p-box representation and applying them to the fine scale model, and then conducting stochastic simulations to obtain a probabilistic response.

4.4.1 Elastic Modulus Sampling

Because usually very little samples are available for scanning, elastic modulus is considered as epistemic uncertainty in this framework. Therefore it will be represented with an interval. In order to obtain an interval from the p-box representation, first *n* values are randomly chosen in the range from 0 to 1 via MCS or LHS. For each of those values, different intervals can be generated from the p-box representation. An example of this with 10 intervals generated via LHS can be shown in Figure 4-5. Next, the fine scale model of the part must be separated into sections, where each section contains a different elastic modulus. This can be done by taking large sections of the part or by considering each individual element as having a different elastic modulus. Elastic modulus values are then sampled from a generated interval using LHS or MCS with a uniform distribution inside the elastic modulus distribution.

4.4.2 Fine Scale Simulations

The response of the fine scale simulations are obtained in a very specific way. For a single generated interval of elastic modulus values, the different sections of the part are assigned different elastic modulus values. Then many simulations are conducted keeping the same elastic modulus values, but modifying the geometry in the same way demonstrated in chapter 3. Thus after many simulations, a single distribution of the desired response can be obtained. Then, a second interval for elastic modulus can be generated and used to apply new elastic modulus values to the part. Once again stochastic simulations are conducted with random geometry until another distribution of the response is obtained.

This process can be repeated until there are a sufficient number of distributions to obtain a p-box representation of the response.

4.5 Coarse Scale Model Construction

The coarse scale model construction uses the same process as section 3.3.1 to obtain the geometry of the coarse scale model, but there are differences when introducing input variables. First, as will be explained in more detail in section 4.6, stochastic upscaling with p-box information is a multi-objective optimization problem. Therefore it is often useful to introduce extra input variables into the coarse scale model. Introducing more input variables allows the optimizer to have more options when attempting to find a solution. In addition, it is important that at least one of the input variables is epistemic and one is aleatory. The aleatory inputs will continue to be represented via PCE, but the epistemic inputs will simply be represented by an interval. This way, simulations can be conducted in the same way as the fine scale simulations to obtain a p-box results.

4.6 Stochastic Upscaling with P-Box

The purpose of the stochastic upscaling with p-box is to find a distribution of input parameters for the coarse scale model such that its response matches that of the fine scale model. In this case however, the response is defined as a p-box. Therefore, for every iteration of the optimization process, the optimizer must go through the process of generating a p-box. That is, epistemic inputs must be generated based on the given interval, aleatory simulations must be performed to produce a single distribution, and then the process must repeat until a p-box can be formed. Due to this high computational cost, it is very important to use surrogate modelling with the coarse scale model, similar to section

3.4.2. Any kind of multi-variable fitting can be used to complete the surrogate modelling, such as multivariable kriging, multivariable polynomial fitting, etc. It is not feasible to waste time performing near identical coarse scale simulations when so many simulations are necessary.

Although the stochastic upscaling processes used before can be applied to multi-objective problems, stochastic upscaling with p-box information is inherently multi-objective. An overview of the stochastic upscaling method with p-box can be seen in Figure 4-6.

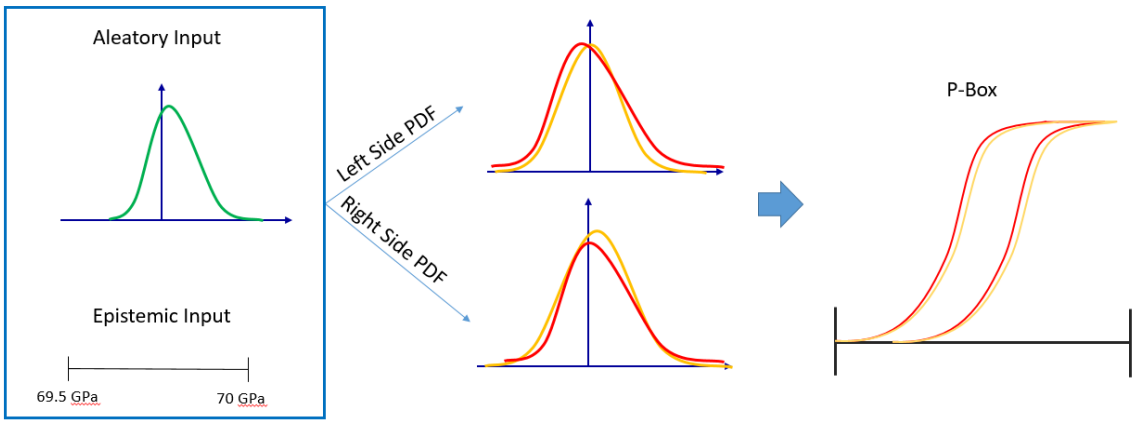


Figure 4-6: Upscaling with P-Box Method

After producing a p-box with the coarse scale model, the left and right boundaries are treated as separate distributions. They can each be compared to the left and right boundaries of the p-box from the fine scale simulations using the exponential loss objective function, and the values can be optimized as separate objectives using any available multi-objective optimization method.

CHAPTER 5. CASE STUDY – TURBINE BLADE EXAMPLE

In order to show its efficacy, the designed framework has been applied to a turbine blade example. This chapter will be organized into 3 main sections. The first main section will cover the geometric uncertainty quantification process of the turbine blade as laid out in chapter 3. The second main section will cover the material uncertainty quantification process of the turbine blade as laid out in chapter 4. Finally, the last section will discuss the validation of the results. The turbine blade used to test the efficacy of the model is based on a Rolls Royce Pegasus turbofan engine design [59]. A CAD model of the turbine blade can be seen in Figure 5-1a.

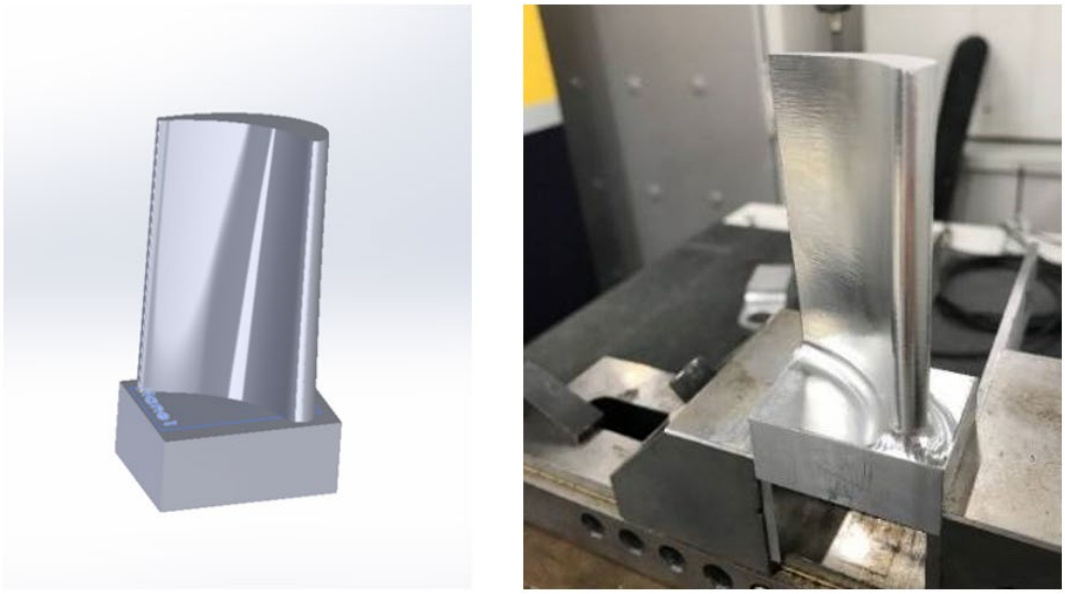


Figure 5-1: Specimen Used in Case Study - a) CAD Model b) Manufactured Part

5.1 Geometry UQ Framework

This section will cover how the geometry UQ framework was applied to the turbine blade example. Therefore the organization of this section will follow the sub-sections of chapter 3. Typically turbine blades are casted out of steel, aluminum, nickel-cobalt, or other similar alloy. Due to not having access as a reliable manufacturing method, the turbine blade was machined instead, as shown in Figure 5-1b. Thus the geometry uncertainty quantification will only apply to the machining manufacturing process. The turbine blade was constructed according to the CAD model in Figure 5-1 out of aluminum.

5.1.1 Geometric Uncertainty Quantification

The turbine blade geometry was measured using a Brown and Sharpe CMM machine as shown in Figure 5-2.

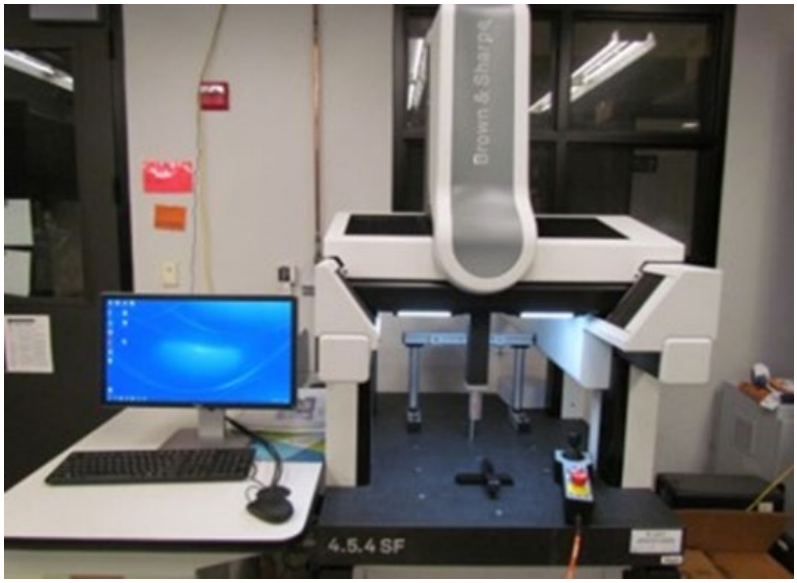


Figure 5-2: CMM Machine Used to Analyze Part

In order to segment the CMM machine recordings, the blade model was broken up into different line segments, where each line segment is a 2D profile of the turbine blade, as shown in Figure 5-3a. There are a total of 46 curves. Along each curve, approximately 200 points were recorded. At each point, information on the point corresponding to the CAD model, the point corresponding to the manufactured part, and the error between the two. It was found during analyzing the CMM data that some data points were recorded obviously incorrectly, as shown in Figure 5-3b. These curves were thrown out of the geometric uncertainty quantification.

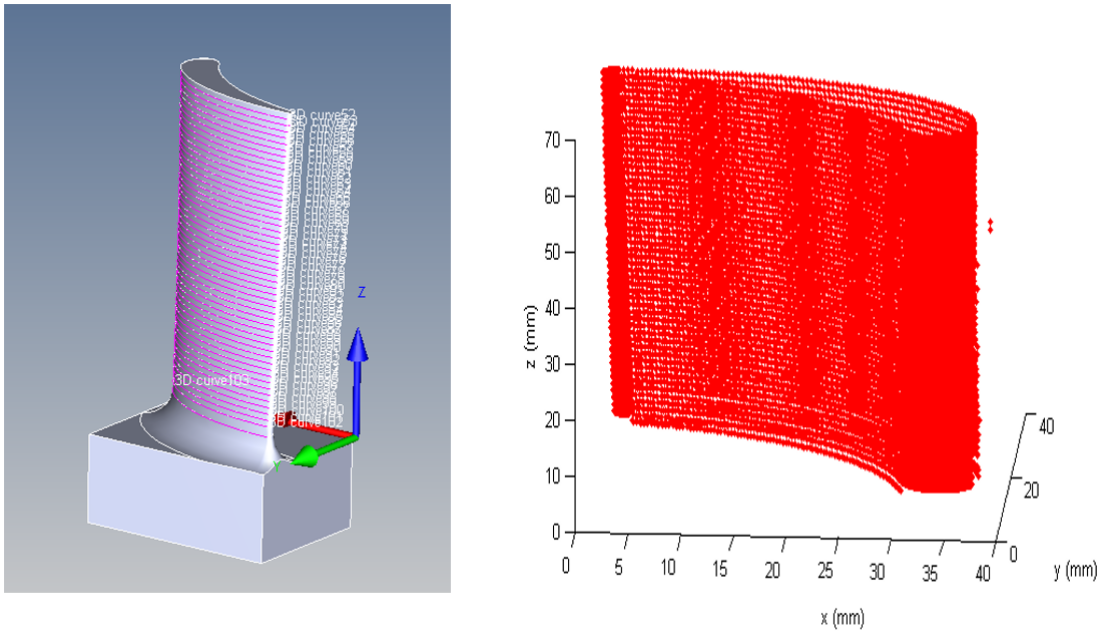


Figure 5-3: Curves Analyzed by CMM Machine - a) Curves on CAD Part b) CMM Measurement Results

Using semivariogram analysis, each curve from Figure 5-3b was analyzed separately. Due to the machining process, each of the curves was considered independent

from each other. When the part is machined, the part is shaped one layer at a time. Therefore, two points next to each other in the z-direction are machined at vastly different times. In addition, when performing initial semivariogram analysis, it was found that 4 different sections of the blade had significantly different semivariograms, as shown in Figure 5-4a. Therefore semivariogram analysis was performed separately for each curve and for each section of the turbine blade. These sections are shown in Figure 5-4b, and will be called concave, convex, small edge, and large edge.

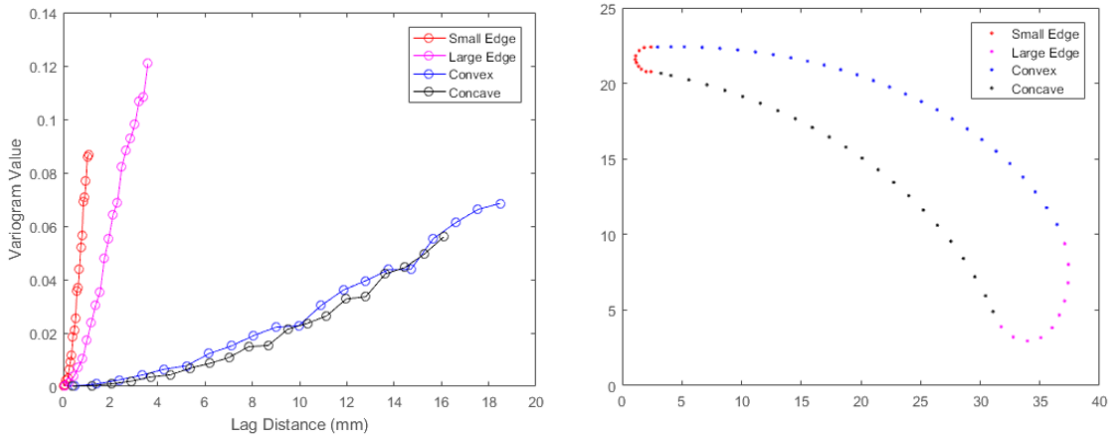


Figure 5-4: Profile Separation - a) Semivariogram Analysis b) Profile Sections

Semivariogram analysis was performed using equation 2, and fitted using the least squares regression method shown in equation 3. An example of the results of the semivariogram analysis for the concave section can be seen in Figure 5-5.

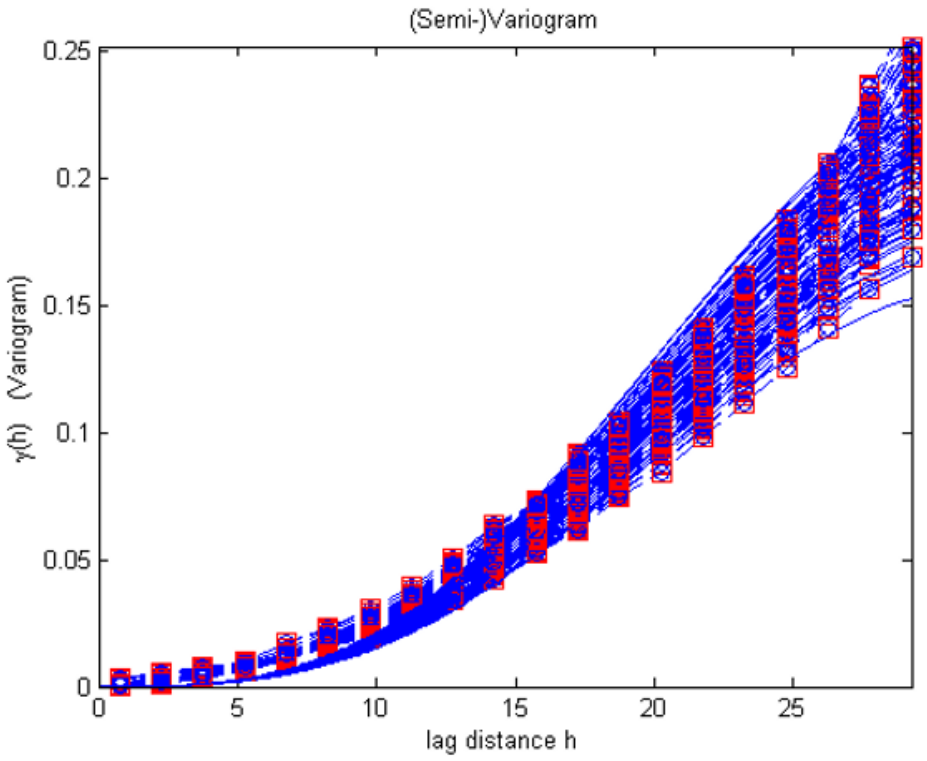


Figure 5-5: Full Semivariogram Analysis of Concave Section

5.1.2 Fine Scale Simulations

The fine scale model of the turbine blade is a finite element model constructed in Abaqus FEA software. The goal is to implement the semivariogram analysis data into simulations.

5.1.2.1 Fine Scale Model Construction

In order to have the fine scale model be an accurate representation of reality, it was modelled based off of the CMM data. Using the points obtained from the CMM machine, 2D profiles were recreated in Solidworks. Then a solid was created by lofting through each of the obtained profiles. The part was then imported into Abaqus FEA software and meshed

using the native meshing tools. The meshed model can be seen in Figure 5-6a. The final model contains 100,172 elements.

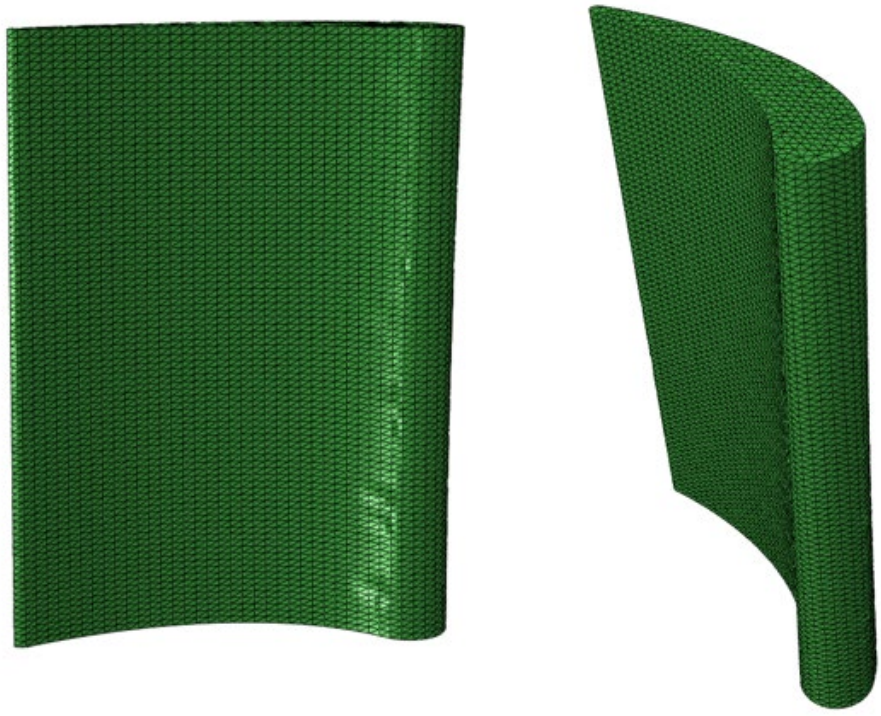


Figure 5-6: Fine Scale Finite Element Model

The parameters of the simulations are an analysis of the natural frequency of the turbine blade with a free-free boundary condition. Natural frequency is often used for analysis of failure of turbine blades. If the natural frequency falls below a specific value, it is considered to have failed [60]. For each simulation, the first 16 natural frequencies were recorded. This was done in an effort to record 10 useful natural frequencies for any possible upscaling in the future. In a free-free boundary condition, the first 6 natural frequencies obtained will be close to 0, as they represent the 6 degrees of freedom allowed for the part. Therefore 16 total frequencies were recorded.

5.1.2.2 Random Geometry Generation

The random geometry of the turbine blade was created using K-L expansion. Since each profile of the turbine blade was taken to be independent of each other, turbine blade profiles were generated separately from each other. In addition, the geometry of the four sections, concave, convex, large edge, and small edge, were each randomized individually. To generate a random profile, first the appropriate semivariogram data (matching the profile location) is used to generate a covariance matrix. Then an eigenvalue decomposition is conducted and K-L expansion is used to generate a set of correlated random errors for the turbine blade. In equation 5, the set of uncorrelated random variables (represented by $\xi(\theta)$) is chosen to be the standard normal Gaussian distribution, since the data is expected to be normal in nature. An example of different turbine blade profiles generated from the same set of semivariogram data can be seen in Figure 5-7.

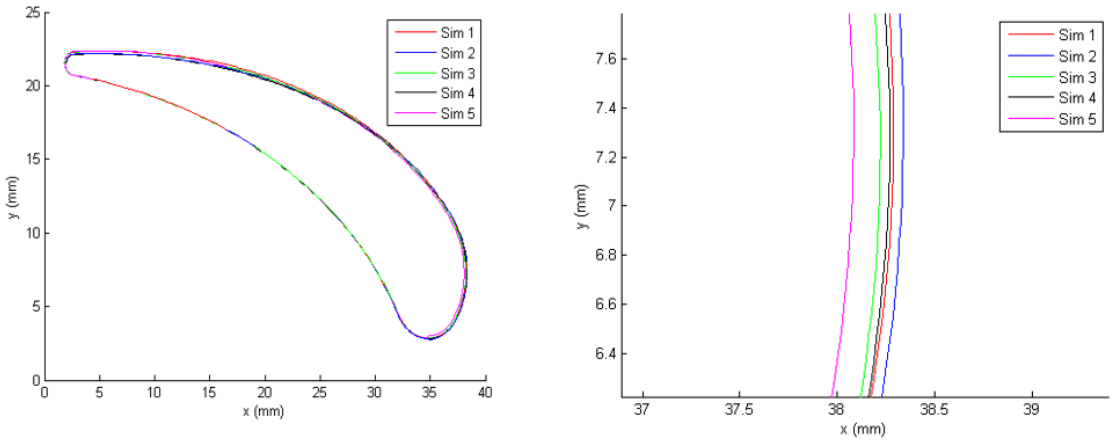


Figure 5-7: Random Blade Profiles - a) Overall View b) Zoomed View of Large Edge

5.1.2.3 Stochastic Simulation

In order to conduct stochastic simulations, the finite element model of the turbine blade created in Abaqus was converted to a .inp file, i.e. an Abaqus input text file. In addition, a list of surface nodes for the finite element mesh was recorded. Each of the surface nodes were grouped together according to their z-coordinates. In this way profiles could be identified from the surface nodes. Each of these profiles were assigned different semivariogram data, depending upon which profile was closest in z-position (according to the profiles in Figure 5-3. After assigning the appropriate semivariogram data, the profiles were also separated into the 4 sections identified as independent from each other: convex, concave, large edge, and small edge. Each of the points in the profile are then randomized according to the K-L expansion described in section 5.1.2.2. The coordinates in the Abaqus input file were modified and the simulation was conducted. It was found that modifying the surface coordinates of the finite element mesh did not produce any poorly shaped elements. Two different turbine blade realization and the contour map showing the differences can be seen in Figure 5-8

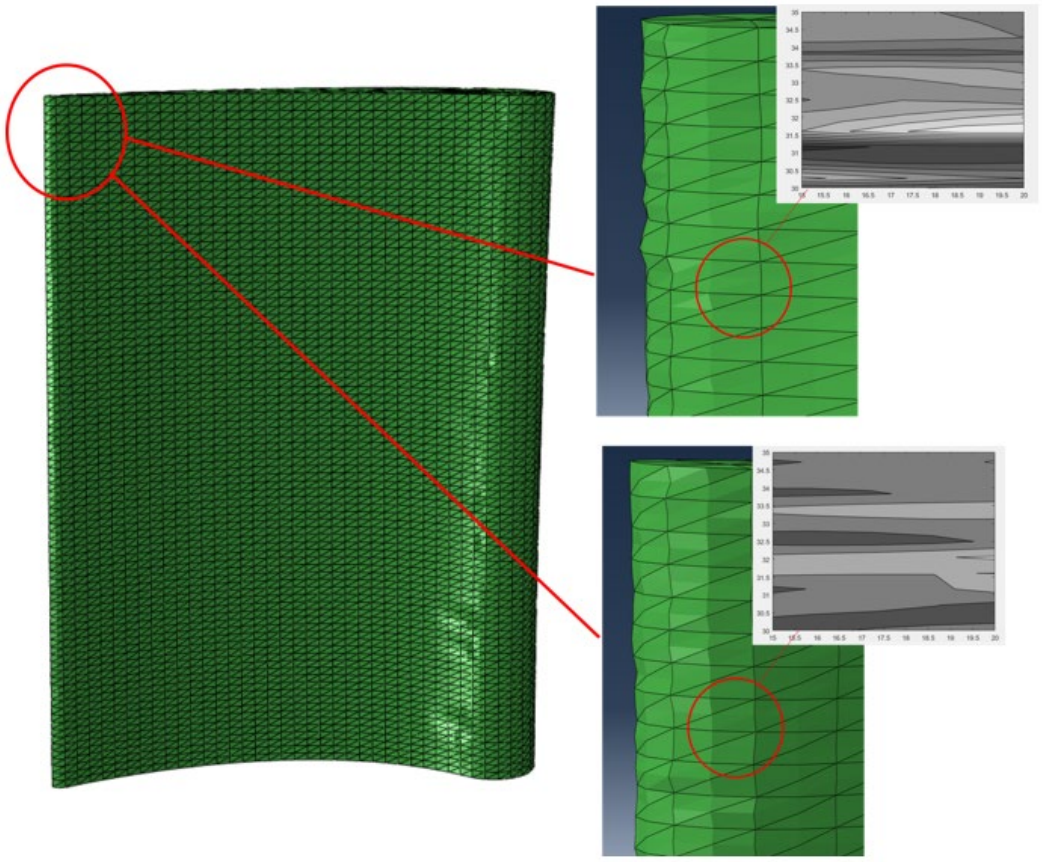


Figure 5-8: Random Surface Modeling Integrated into Fine Scale Finite Element Model

200 simulations were performed in this fashion, and the first natural frequency was recorded for each. A PDF of the results can be seen in Figure 5-9.

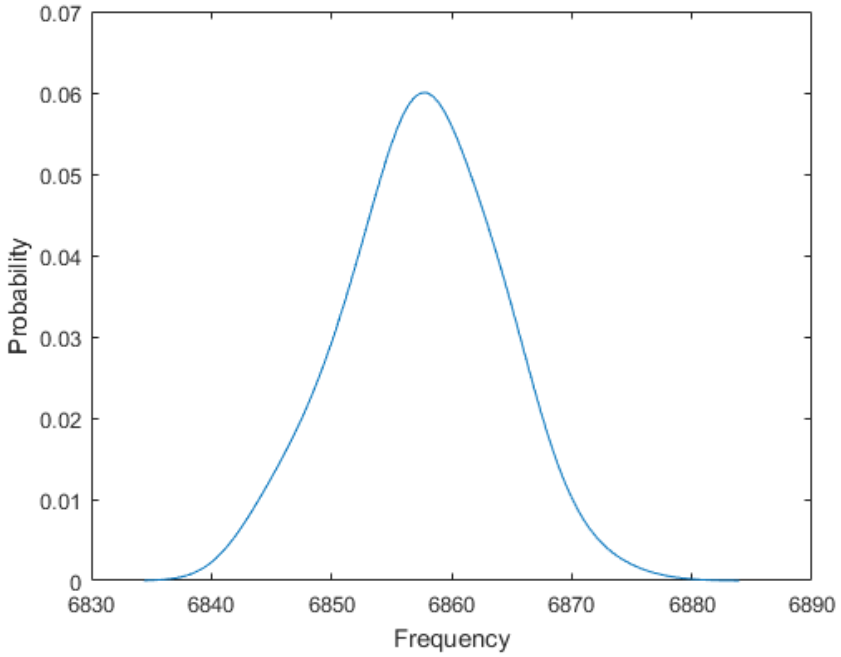


Figure 5-9: PDF of Fine Scale Simulation Results

5.1.3 Coarse Scale Model Construction

The next step is to create a coarse scale model that approximates the geometry of the fine scale model. This was done via Bezier curves and Bezier surfaces.

5.1.3.1 Surface Approximation

In order to approximate the surface of the fine scale model of the turbine blade, a method of generating several profiles and then lofting through the profiles to generate the surface was used. In order to determine the placement and number of these profiles, the semivariogram analysis was used. The semivariograms for each of the profiles were analysed for similarity, and put into groups based on the similarity. An example of the sets

of semivariograms separated into different groups can be seen in Figure 5-10a. These 5 different sections as shown on the turbine blade in Figure 5-10b.

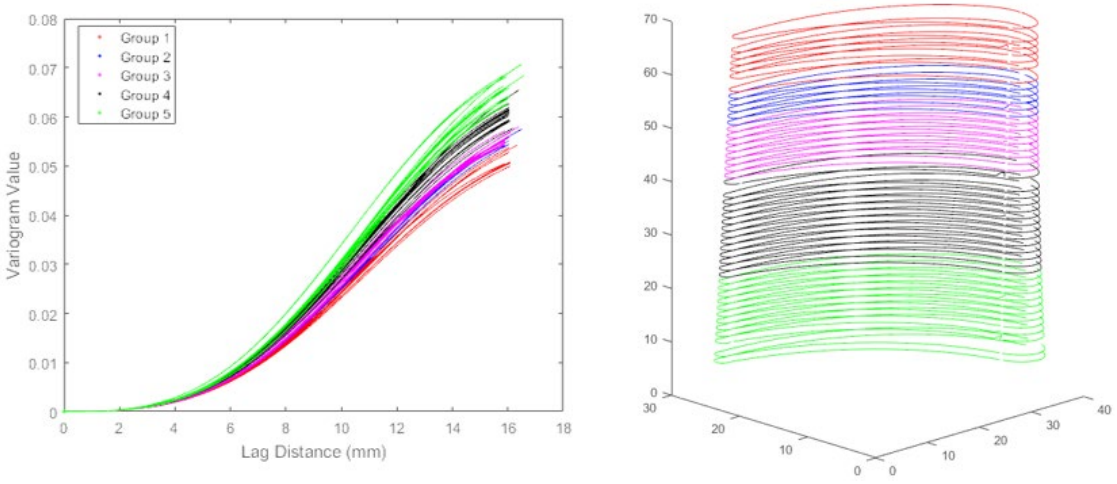


Figure 5-10: Turbine Blade Section Creation - a) Variogram Analysis b) Turbine Blade Sections

The two end profiles, as well as the profiles separating the different sections were used as the profiles to be modelled by Bezier curves. Each of the profiles were represented by 4 Bezier curve, where each curve represents a different section: concave, convex, large edge, and small edge. This leads to a total of 20 Bezier curves used to represent the geometry of the turbine blade. Each Bezier curve was a 3rd degree Bezier curve and the coefficients were determined using least squares residual analysis with the points recorded from the CMM machine. After computing each of the Bezier curves, 3x5th degree Bezier surfaces were used to loft the surface along the turbine blade profiles. This process can be seen in Figure 5-11

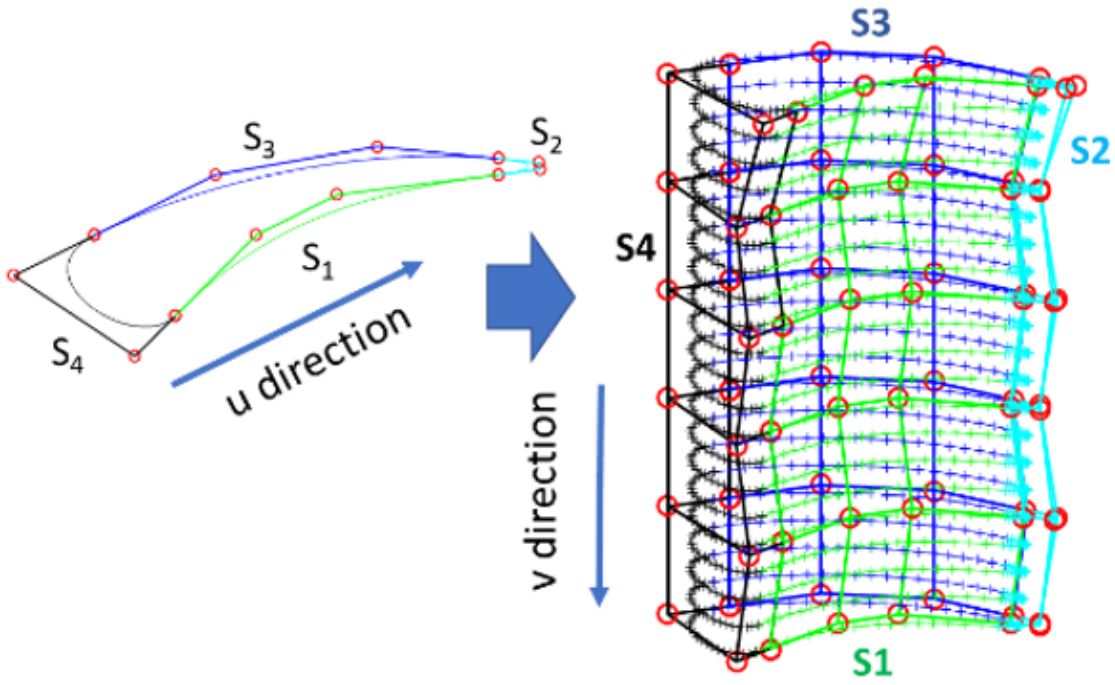


Figure 5-11: Bezier Surace Creation Process

5.1.3.2 Finite Element Model Creation

After estimating the surface geometry of the turbine blade, a finite element mesh must be generated for the blade. An overview of this process can be seen in Figure 5-12. The first step in this process is to create a surface mesh for the turbine blade. This was done by generating a grid of points for u and v with values between 0 and 1 for each Bezier surface. After plugging these values into the Bezier surface equations, a grid of points can be created for the lateral surfaces of the turbine blade, as shown in Figure 5-12a.

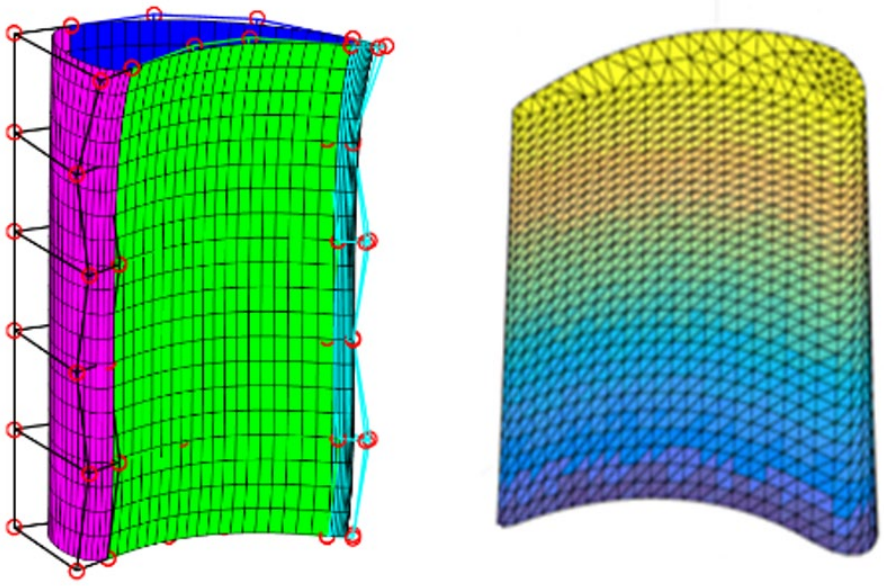


Figure 5-12: Surface Mesh Creation - a) Bezier Surface Grid Points b) Final Surface Mesh

Then each Bezier surface was meshed using Delaunay triangulation, keeping sure that the orientation of the mesh pointed towards the outside of the turbine blade. Next, points were added to the top and bottom of the turbine blade to create a surface mesh for those surfaces. This was done by systematically adding points, trying to keep the distance between each of the points similar to the length of the lines in the mesh created with Delaunay triangulation. Then the top and bottom surfaces were meshed using Delaunay triangulation again. The result can be seen in Figure 5-12b.

Finally, points were added to the inside of the turbine blade and meshed using 3D Delaunay triangulation. The inside of the turbine blade was determined using the mesh orientation of each of the surfaces. Therefore ensuring correct mesh orientation with 2D Delaunay triangulation is critical. This volumetric meshing process was done using *tetgen*,

a previously developed software [61]. The final coarse scale finite element model contains approximately 900 elements. Since the mesh is iteratively generated, the mesh can contain a slightly different number of elements each time. After creating the finite element model, it was imported into Abaqus and the same simulation parameters were applied, as shown in Figure 5-13.

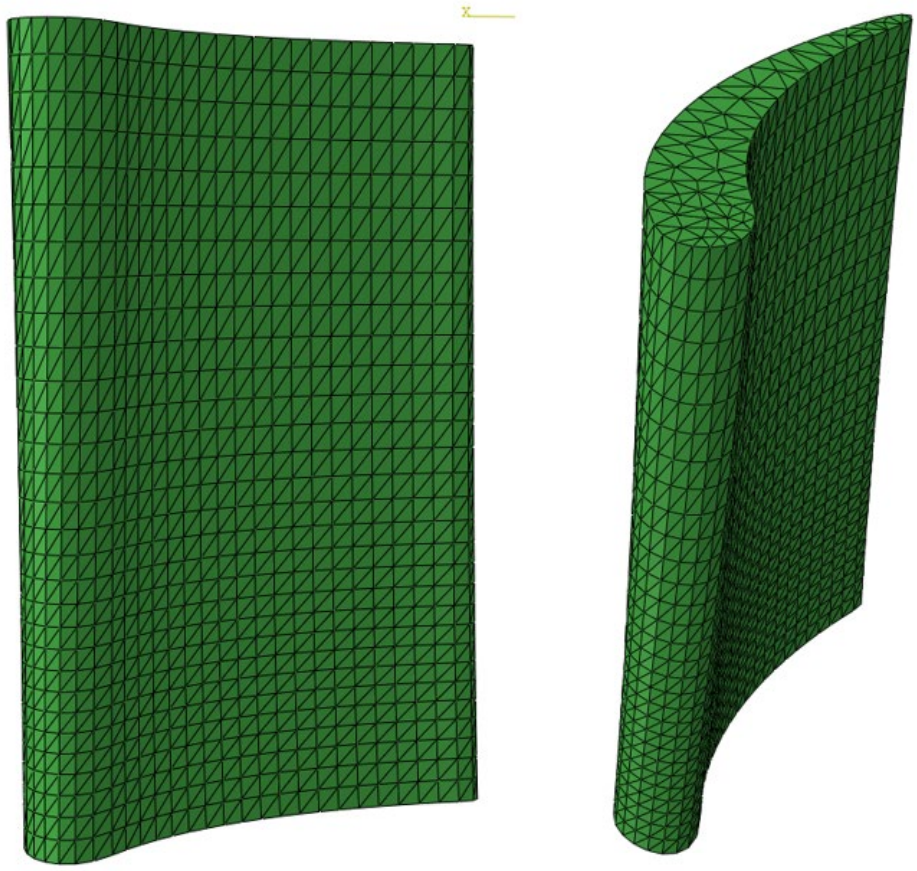


Figure 5-13: Coarse Scale Finite Element Model

5.1.4 Stochastic Upscaling with One Output

The purpose of the stochastic upscaling method in this example is to match the first natural frequency of the coarse scale model to the response obtained from the fine scale

model, as shown in Figure 5-13. This is done by first introducing an input variable, constructing a surrogate model, and performing optimization to determine the distribution of the input variable.

5.1.4.1 Input Variable

Due to the method of construction, it is possible to use any of the Bezier surface control points as input variables for the optimization. However, since the optimization is only for one output, a simpler input variable was chosen. For this optimization, a scaling factor, called Δ was introduced. This represents the overall size of the coarse scale model, i.e. it modifies all of the Bezier surface control points simultaneously. Here a scaling factor value of 1.0 is a coarse scale model that is the same size as the fine scale model, a scaling factor of 1.2 represents a coarse scale model that is 20% larger, and a scaling factor of 0.8 represents a coarse scale model that is 20% smaller.

5.1.4.2 Surrogate Modelling

In order to reduce the computational effort of the stochastic upscaling, a surrogate model was introduced to model the coarse scale model. A series of points were created using the coarse scale model relating the first natural frequency to the scaling factor, Δ. A range of 0.4 to 1.6 was used for Δ with a total of 500 values. Several different methods were used to attempt to fit the data, including Kriging, polynomial fits, and others. The results of the fitting process can be seen in Figure 5-14.

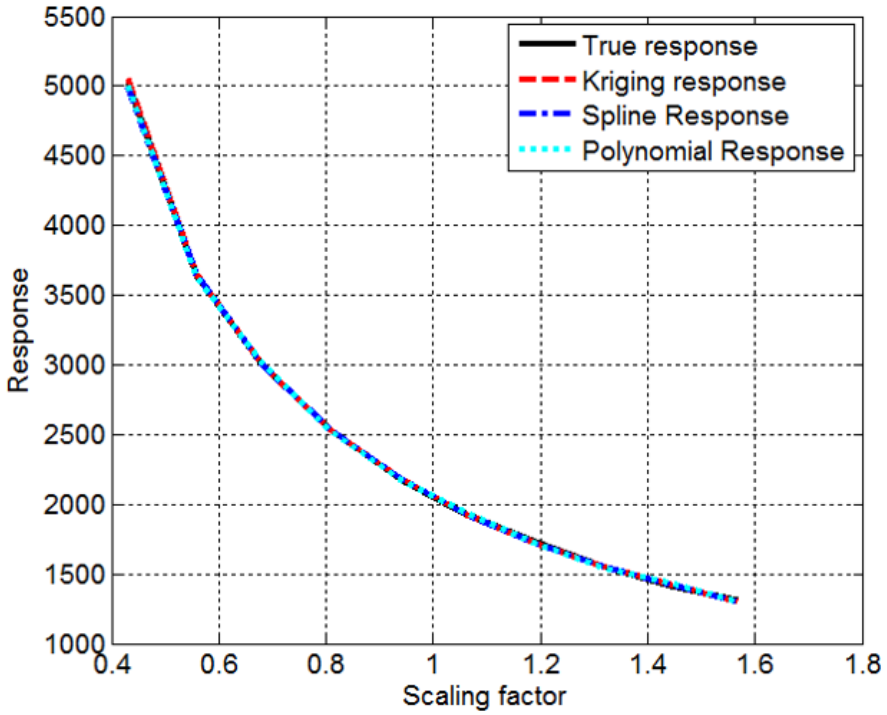


Figure 5-14: Curve Fitting for Surrogate Model

All of the surrogate modelling methods produced good results, however Kriging was chosen as the surrogate modelling method because of its common use with uncertainty related situations [62][63].

5.1.4.3 Optimization

The input variable, Δ was modelled using PCE. Although a 3rd order PCE was eventually used, it was often useful to assume that the latter 2 coefficients of PCE were small and attempt to model Δ using a 1st order PCE. After completing the optimization once, the result was used to set strict bounds on an optimization with 3rd order PCE. The optimization was completed first through a genetic algorithm. After obtaining a solution

through the genetic algorithm, a derivative based optimization approach was used to arrive at a final solution. The final coefficients for PCE can be seen in Table 2.

Table 2: Polynomial Chaos Expansion Coefficients Used to Represent Δ

	b_0	b_1	b_2	b_3
Value	1.273	-3.872e-4	-1.512e-5	-1.492e-5

The distributions of first natural frequency for both the fine and coarse scale models can be seen in Figure 5-15.

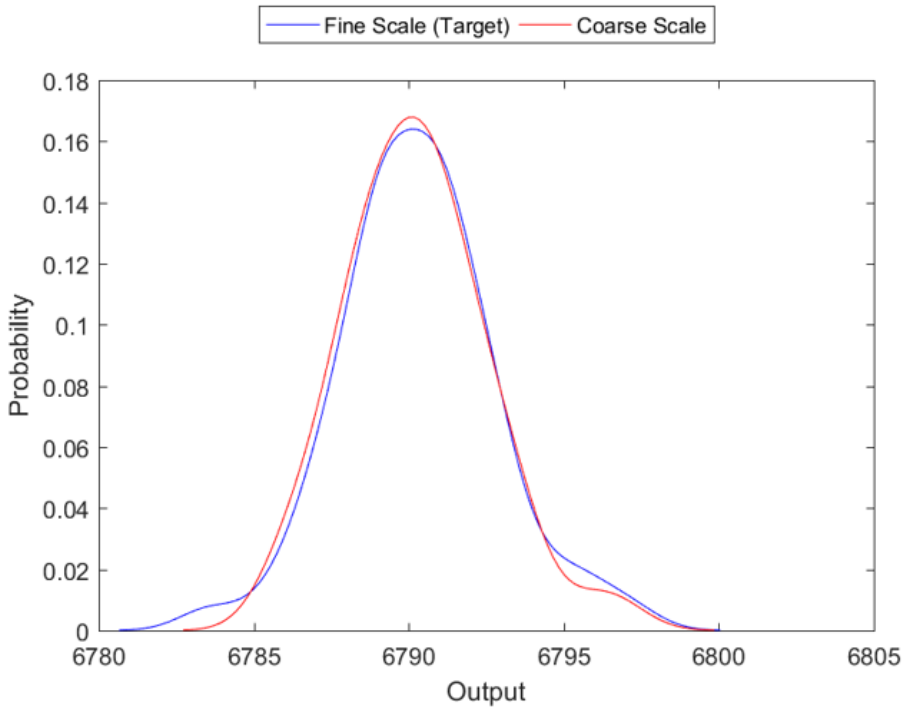


Figure 5-15: PDF of First Natural Frequency for Coarse and Fine Scale Models

5.1.5 Stochastic Upscaling with Multiple Outputs

The upscaling process with only aleatory uncertainty was attempted with multiple outputs. Mild success was seen with two outputs, however using 3 or more outputs resulted in the optimizer being unable to find a solution. Therefore only the results for 2 outputs are demonstrated here. The coarse scale model was upscaled in order to match the results for the first and second natural frequency.

5.1.5.1 Input Variable

Because of increased complications related to having multiple objectives, an extra input variable was introduced into the model. Instead of using a single scaling factor, Δ, 2 scaling factors were used, namely Δx and Δy. In this case, Δx represents the scaling factor in the x-direction while Δy represents the scaling factor in the y-direction. There is no Δz value introduced therefore the z-coordinates of all of the Bezier control points remain constant.

5.1.5.2 Surrogate Modelling

Due to having multiple inputs, creating a surrogate model becomes much more complicated. In order to deal with this problem multivariable surface fitting methods were used to create the surrogate modelling. Specifically, in this case, a multivariable polynomial fit was used. The power of the polynomial fit was slowly increased and the error plots were compared until the least error was found. A 2^{nd} degree polynomial was used and the error plots can be seen in Figure 5-16.

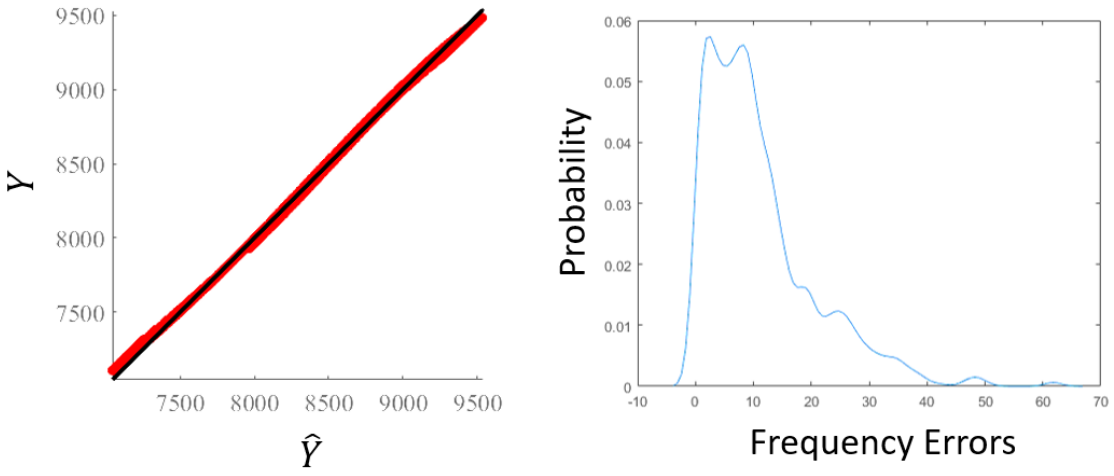


Figure 5-16: Error Plots for Polynomial Fit - a) Goodness of Fit Plot b) Error PDF

5.1.5.3 Optimization

PCE was used to represent both Δx and Δy. Similar to section 5.1.4, a 1st order PCE was initially solved and used to set bounds on optimization with a 4th order PCE. A multi-objective genetic algorithm was used to match the distributions of the first and second natural frequency. The final coefficients of PCE can be seen in Table 3.

Table 3: Polynomial Chaos Expansion Coefficients Used to Represent Δx and Δy

	b_0	b_1	b_2	b_3	b_4
Δx	1.966	4.510e-3	3.397e-4	-9.554e-5	8.883e-5
Δy	1.319	8.348e-3	8.256e-4	-1.032e-4	2.107e-4

The results of the upscaling process can be seen in Figure 5-17.

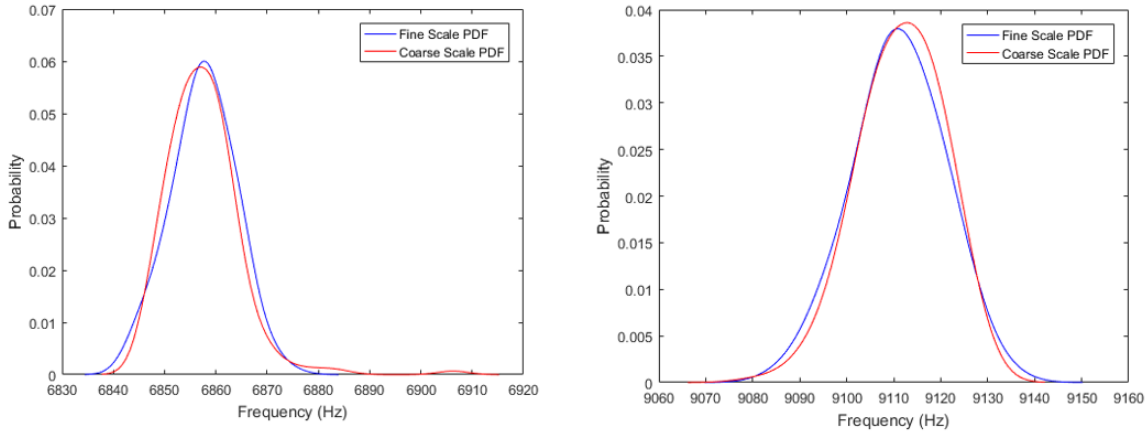


Figure 5-17: PDF's of First and Second Natural Frequencies of Coarse and Fine Scale Models

5.2 Material UQ Framework

The purpose of this section is to show how the material UQ framework was applied to the turbine blade example. Therefore the organization of this section will follow the sub-sections of chapter 4. Samples of material were analyzed using a CT scanner, and fine scale simulations were conducted using the porosity information. Upscaling was performed to obtain information on elastic modulus, and this was applied to the fine scale finite element model from section 5.1. Finally, upscaling was conducted to obtain a finalized coarse scale model.

5.2.1 Analyzing CT Scan Data

A total of 4 samples of material was analyzed using a Zeiss CT scanner. For each of the 4 samples, a set of 2D images were generated representing the inside of the material.

For each set of 2D images, part of the area contained information on the inside of the material, and part of the area contained the empty space surrounding the sample of material. In order to make analysis easier, the surrounding portions of images containing empty space were cut off. This was done so that each section contained the same number of pixels. The number of pixels for each cropped set of 2D images as well as the resolution of the images can be seen in Table 4. Here, the resolution is the size of the side of the pixel (represented by cubes).

Table 4: CT Scan Data Information

	Pixels	Resolution
Values	283x364x184	12.74 μm

5.2.1.1 Interpreting CT Scan Data

The images were filtered using multiple methods, but it was found that using several rounds of Gaussian filtering resulted in the most realistic results. Specifically, 3 rounds of Gaussian filtering with a block size of 3 was used. This was decided based on visual inspection of the results as well as comparing the results to existing CT scan images of the material [35]. An example of the images before and after filtering can be seen in Figure 5-18.

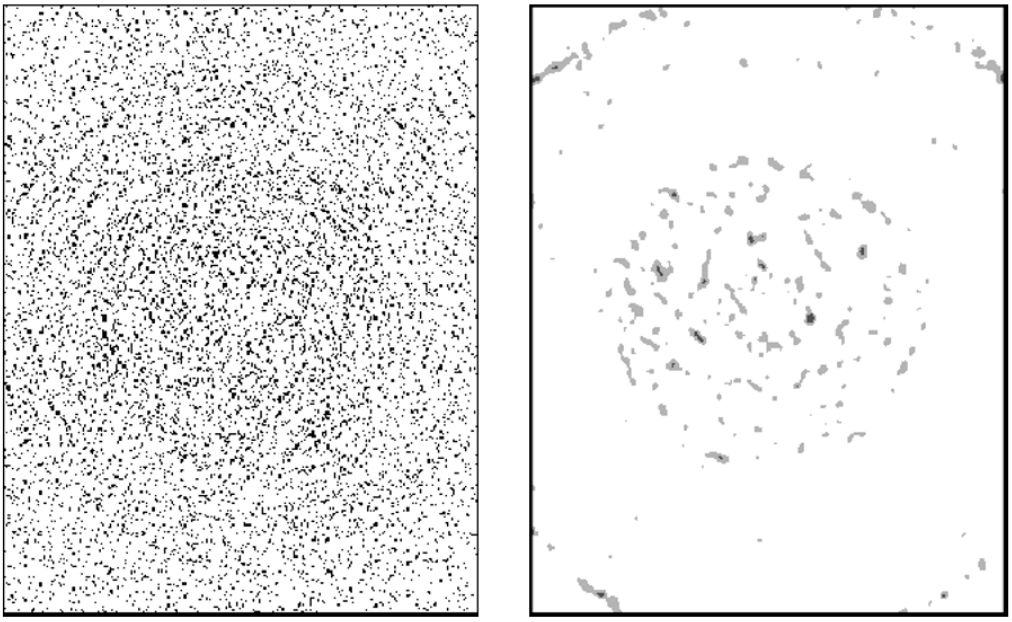


Figure 5-18: CT Scan Images - a) Unfiltered Image b) Filtered Image

After filtering, the images were segmented into binary images using a cut-off value. This cut-off value was decided on using visual inspection and a mean method. Specifically, a maximum and minimum greyscale value were chosen. Anything below the minimum was set to the minimum value, and anything above the maximum value was set to the maximum value. Then the data was re-scaled so that minimum value was 0 and the maximum value was 255. This allowed for easier distinction between pixels containing pores and pixels containing material. Afterwards, the images were segmented into a binary image using a mean value of 127.5 as the cut-off value.

5.2.1.2 Extracting Porosity Information

After filtering and segmenting the images, the next step is to create a voxel model of the pores. Since the images were cropped so that no empty space surrounding the

samples was contained in the images, the voxel models were exported so that voxels represented pores instead of material. In addition, information was recorded on the overall size of each sample. In this way, the voxel model could be exported with voxels representing without losing any information. This allowed for much quicker analysis. An example an exported voxel model from one of the samples can be seen in Figure 5-19. The voxel models were exported as an Abaqus input file and viewed using Abaqus FEA software.

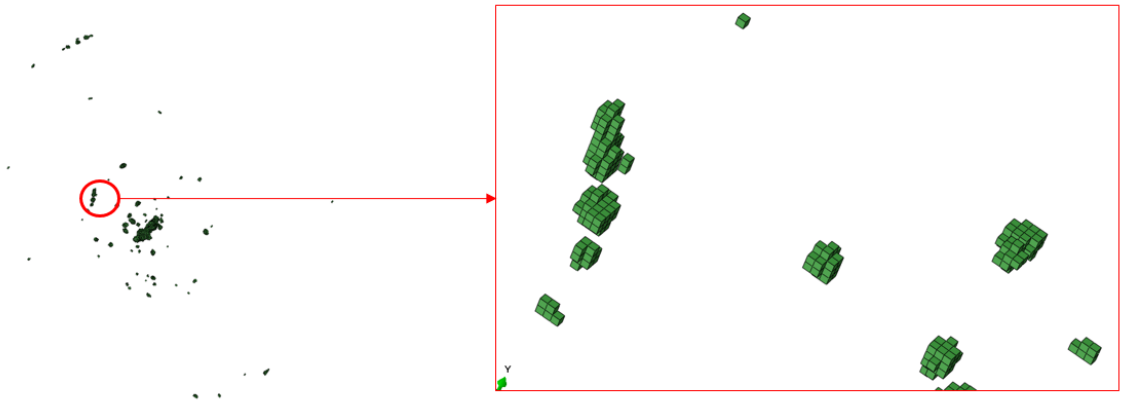


Figure 5-19: Voxel Model of Pores

5.2.2 Fine Scale Model Construction

Next, fine scale models were constructed using the information obtained from section 5.2.1. Beam models of dimensions 20x20x100 voxels (overall dimension being 254x254x1274 μm) were used for the model. For each sample, a separate set of fine scale simulations were conducted. The beam models were filled with pores whose size was determined by sampling from the pore size distribution of the corresponding sample with MCS. The beam model was filled with pores until the density matched that of the sample.

If a pore would be added that made the density lower than the target density, it was changed to a size that resulted in the closest possible density match. The pores were added to a position in the beam randomly chosen through MCS, avoiding any pores that would appear on the surface. An example of a fine scale model generated in this fashion can be seen in Figure 5-20.

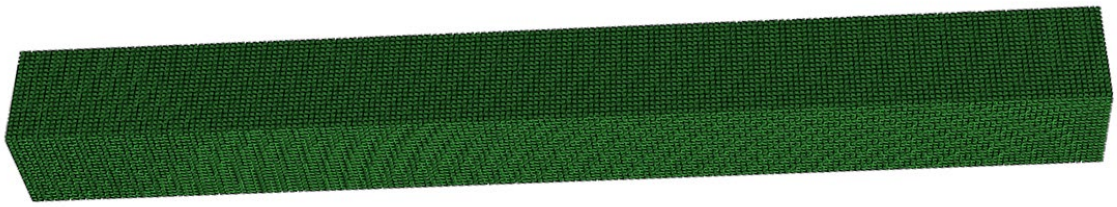


Figure 5-20: Material UQ Fine Scale Model Example

For each sample, to ensure that the eventual elastic modulus was applicable to a large range of strains, 3 sets of simulations were performed, each with a different applied strain: 0.005, 0.01, and 0.015. For each sample, at each applied strain, 200 simulations were conducted for a total of 3000 fine scale simulations. The final result is 3 sets of reactions force graphs, each containing a distribution of reaction force for each sample (4).

5.2.3 Stochastic Upscaling for Elastic Modulus

Next, a coarse scale model was constructed. As laid out in chapter 4, the coarse scale model is just a beam of the same dimensions containing only 1 element. The input for the coarse scale model is the distribution of elastic modulus, and the outputs are the 3 reaction force distributions corresponding to the 3 applied strains of 0.005, 0.01, and 0.015.

The coarse scale model requires no extensive finite element model, as the reaction force can simply be calculated using equation 23.

Next, upscaling was conducted for each material sample. Since 3 different strains were used, a multi-objective stochastic upscaling method was used for optimization. For each material sample, the goal of the optimization was to determine a distribution of elastic modulus such that the distributions of reaction force of the coarse scale model matched those obtained from the fine scale models for each applied strain. An example result of this process can be seen in Figure 5-21.

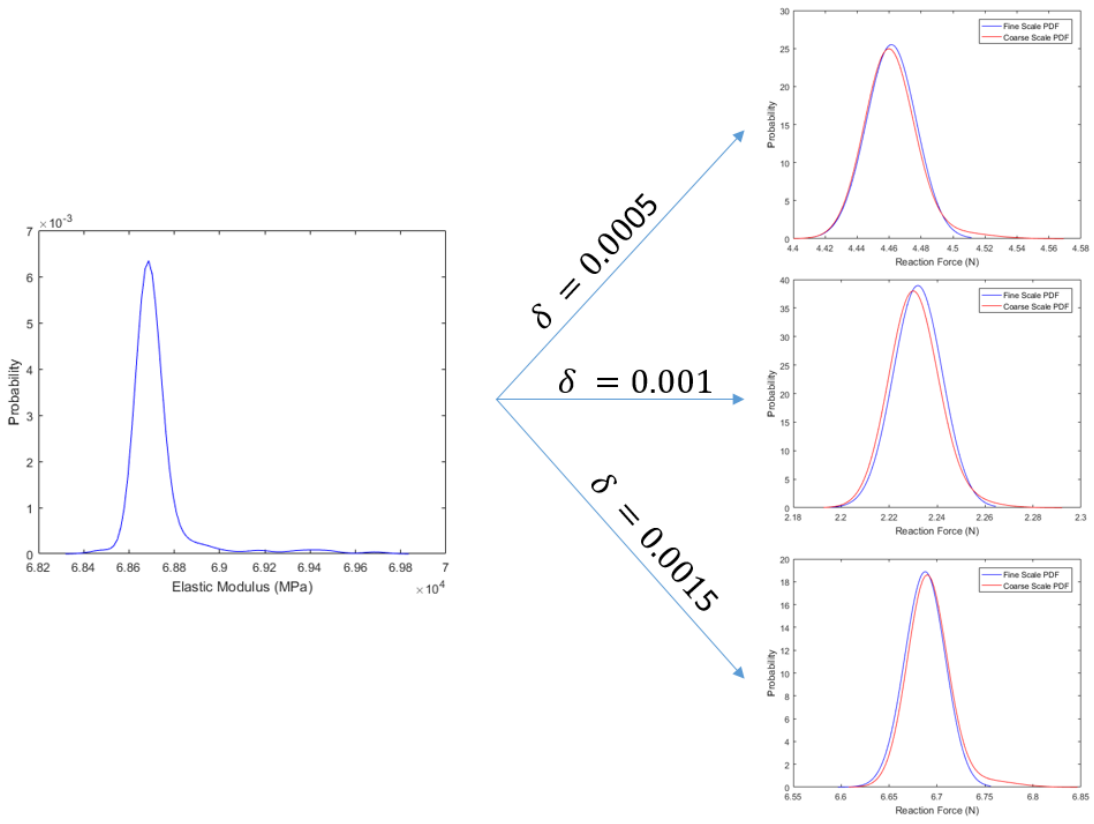


Figure 5-21: Upscaling Example Result for Multiple Strain Values

The elastic modulus was represented using a 3rd order PCE, and it was these coefficients of PCE that were determined in the optimization process. The optimization was performed using a multi-objective genetic algorithm in Matlab. The final coefficients of PCE for each sample can be seen in Table 5.

Table 5 Polynomial Chaos Exapnsion Coefficeints Used to Represent Elastic Modulus Distributions

	b_0	b_1	b_2	b_3
Sample 1	4.463	8.350e-3	9.149e-3	3.414e-3
Sample 2	4.465	7.995e-3	8.692e-3	3.534e-3
Sample 3	4.464	6.715e-3	4.327e-3	4.699e-3
Sample 4	4.462	5.157e-3	5.742e-3	4.625e-3

The final distributions of elastic modulus were constructed using the PCE coefficients. The 4 elastic modulus distributions as well as the resulting P-Box can be seen in Figure 5-22a and Figure 5-22b, respectively.

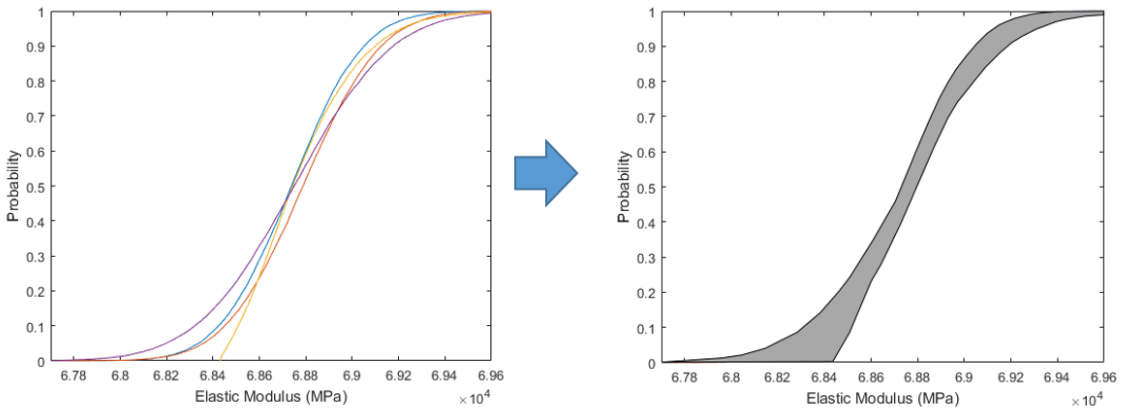


Figure 5-22: Elastic Modulus P-Box - a) 4 Elastic Modulus CDF's b) Elastic Modulus P-Box

5.2.4 Elastic Modulus Uncertainty Propagation

In order to propagate the elastic modulus information obtained in section 5.2.3, the fine scale model from chapter 3 was modified. The fine scale model was broken up into 5 sections, where each section is described by a different elastic modulus. These 5 sections were determined based on the same 5 sections determined from the variogram analysis in section 5.1.1. Earlier, these sections were used in the coarse scale model construction, but here they are used for sections with different elastic moduli. The fine scale model with 5 different sections can be seen in Figure 5-23.

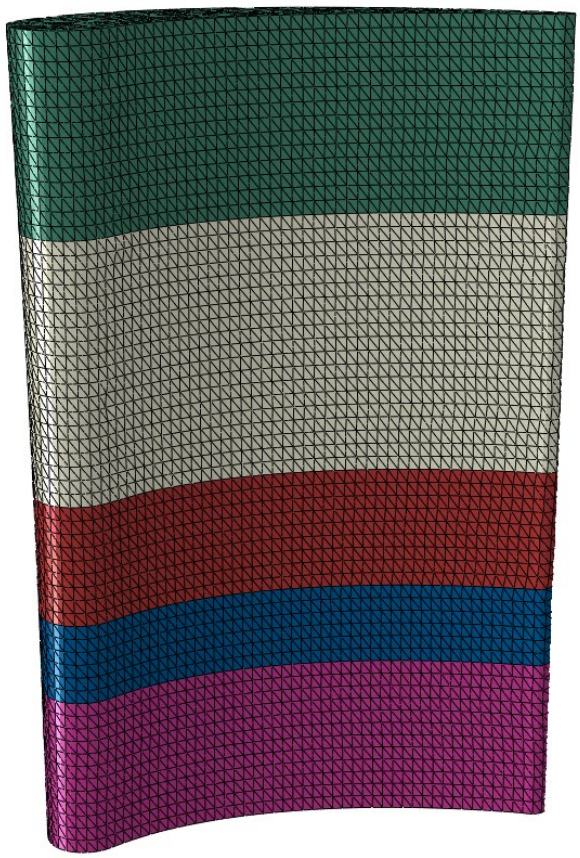


Figure 5-23: Fine Scale Finite Element Model for Aleatory and Epsitemic Uncertainty

In order to conduct fine scale simulations, the method outlined in section 4.4.2 was used. In addition, the elastic moduli were sampled with the same method outlined in section 4.4.1. More specifically, elastic moduli were sampled from the P-Box shown in Figure 5-22. Intervals were randomly sampled from the P-Box, and 5 different elastic moduli were sampled from each interval. After applying the elastic moduli, 200 simulations were conducted where the geometry was changed in the same way described in section 3.2. For each simulation, the same simulation parameters from chapter 3 were used. The boundary

condition was free and the first natural frequency was recorded. The resulting P-Box can be seen in Figure 5-24.

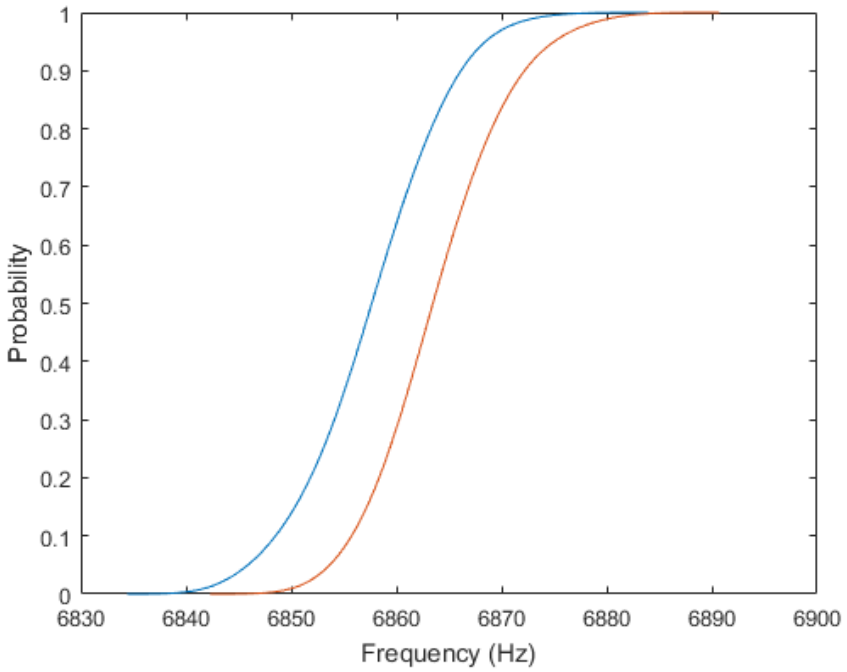


Figure 5-24: P-Box for First Natural Frequency

5.2.5 Coarse Scale Model Construction

In this section, two different coarse scale models were considered. Both were used in separate upscaling process with the goal of identifying a superior coarse scale model in the validation section. The different coarse scale models along with their different inputs will be discussed in separate sections.

5.2.5.1 Coarse Scale Model 1

The geometry for coarse scale model 1 was determined in the same way as the coarse scale model shown in section 5.1.3 and therefore contains a similar number of elements. Model 1 was separated into the same 5 sections as the fine scale model described in section 5.2.4. It can be seen in Figure 5-25.

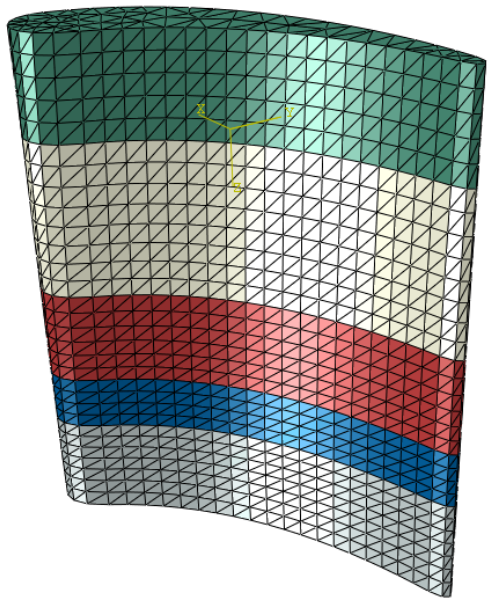


Figure 5-25: Coarse Scale Finite Element Model 1

In this way, there are 5 different elastic moduli inputs, resulting in 5 different epistemic inputs for the model. For aleatory inputs, a single scaling factor, Δ, was used. This scaling factor behaves the same way as the one previously used, where a scaling factor of 1.0 represents a coarse scale model that is the same dimensions (approximately) as the fine scale model. Thus a total of 6 inputs are used: 5 epistemic and 1 aleatory. In upscaling, the

epistemic inputs are represented by an interval, and the aleatory input is represented with PCE.

5.2.5.2 Coarse Scale Model 2

Similar to model 1, the geometry for coarse scale model 2 was determined the same way as the coarse scale model shown in section 5.1.3, and therefore contains a similar number of elements. Coarse scale model 2, however, contains only 1 section, and therefore is represented by a single elastic modulus. Coarse scale model 2 can be seen in Figure 5-26.

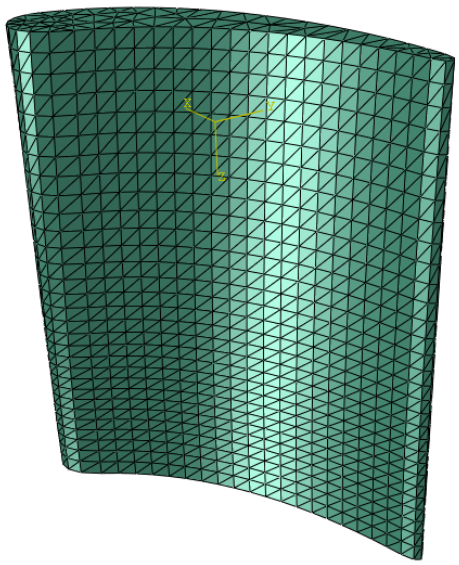


Figure 5-26: Coarse Scale Finite Element Model 2

In order to compensate for the number of inputs (since upscaling with P-Box is multi-objective) 2 aleatory inputs are used. Similar to the coarse scale model used in section 5.1.5. Two scaling factors, Δx and Δy were introduced as aleatory inputs. Thus, a total of

3 inputs are used: 1 epistemic and 2 aleatory. Once again, in upscaling, the epistemic inputs are represented by an interval and the aleatory inputs are represented with PCE.

5.2.5.3 Comparison of Models

Both models contain approximately the same number of elements, therefore each simulations takes a similar amount of time. The largest downside of coarse model 1 is that the surrogate modelling process is extremely computationally expensive. Due to the fact that 6 separate inputs are used, generating a large enough data set requires hundreds of thousands of simulations. On the other hand, the increased number of inputs allows for more control in the upscaling process, therefore possibly making it easier to find a good solution in the multi-objective P-Box upscaling process.

5.2.6 Stochastic Upscaling with P-Box

The goal of this stochastic upscaling process is to determine a set of inputs (for both coarse scale models) such that the resulting P-Box distribution of the coarse scale models matches that of the fine scale model shown in Figure 5-24. Once again, surrogate modelling was used to reduce the amount of time wasted on completing near identical simulations.

5.2.6.1 Surrogate Modelling

Surrogate modelling was completed for both coarse scale models 1 and 2. For both, a multivariable polynomial fit was used. For coarse model 1, a polynomial of order 2 was used. The error graphs for the fit can be seen in Figure 5-27. Higher order polynomial fits showed diminishing returns.

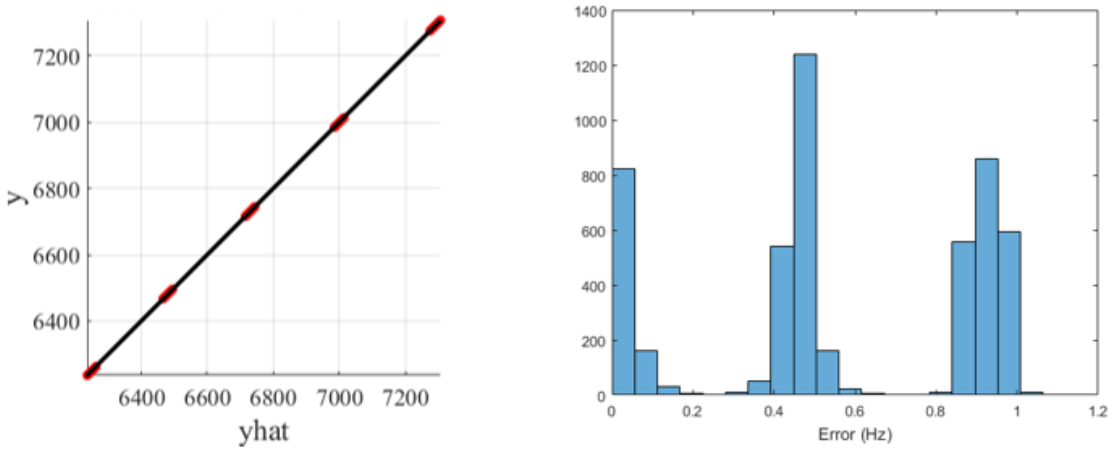


Figure 5-27: Error Plots for Coarse Model 1 - a) Goodness of Fit Plot b) Error Histogram

For coarse model 2, a polynomial of order 3 was used. The error graphs for the fit can be seen in Figure 5-28. Higher order polynomial fit showed diminishing returns.

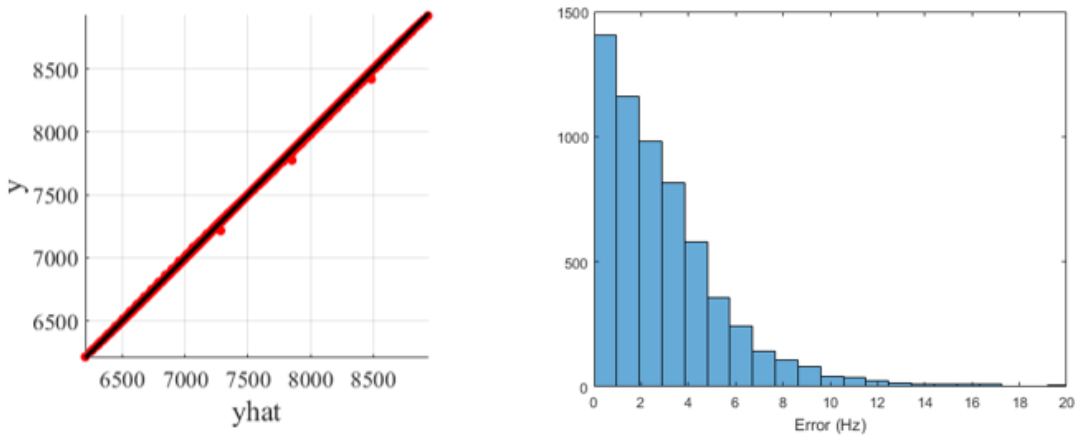


Figure 5-28: Error Plots for Coarse Model 2- a) Goodness of Fit Plot b) Error Histogram

5.2.6.2 Stochastic Upscaling with P-Box

Stochastic upscaling was completed for both coarse models 1 and 2. The optimization was completed with the left and right side of the P-Box being considered as separate objectives. The P-Box for a set of coarse scale simulations is done the same way as described in section 4.6.

Stochastic upscaling with coarse model 1 was completed with a 3rd order PCE. This resulted in 4 variables for the aleatory inputs and 10 variables for epistemic inputs (2 per variable, a maximum and minimum value) for a total of 14 inputs. The coefficients of PCE and intervals for elastic modulus used can be seen in Table 6 and Table 7, respectively.

Table 6: Polynomial Chaos Expansion Coefficients for Scaling Factor of Coarse Model 1

	b_0	b_1	b_2	b_3
Δ	1.276	-1.131e-3	-9.482e-5	-3.352e-5

Table 7: Lower Bound and Upper Bound for Elastic Moduli Values for Coarse Model 1

	Lower Bound	Upper Bound
E_1	6.870e4	6.895e4
E_2	6.890e4	6.896e4
E_3	6.881e4	6.893e4
E_4	6.876e4	6.887e4
E_5	6.860e4	6.882e4

The resulting P-Box distribution of the coarse scale model as compared to the fine scale model can be seen in Figure 5-29. Stochastic upscaling with coarse model 2 was completed with two 3rd order PCE representations. This resulted in 8 variables for aleatory inputs and 2 variables for epistemic inputs for a total of 10 inputs. The coefficients of PCE and intervals for elastic modulus can be seen in Tables Table 8 and Table 9, respectively.

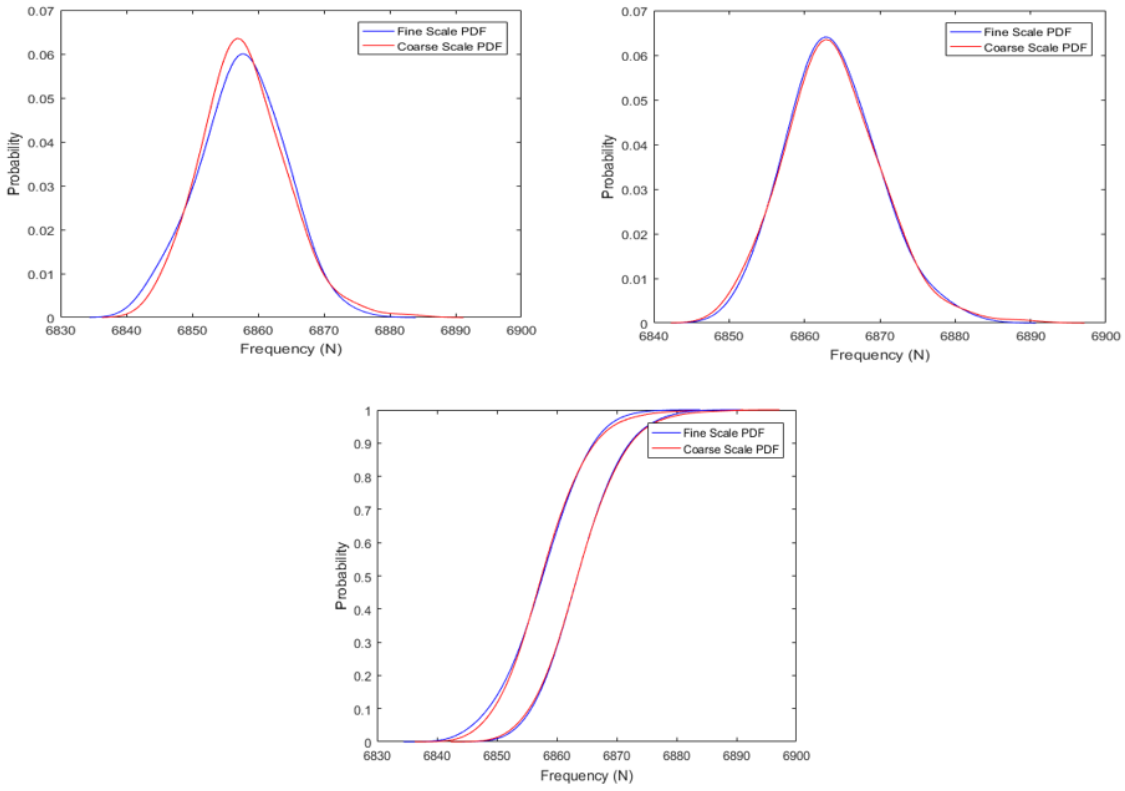


Figure 5-29: Upscaling Results for Coarse Model 1 - a) Left Side PDF b) Right Side PDF c) P-Box

Table 8: Polynomial Chaos Expansion Coefficients for Scaling Factors of Coarse Model 2

	b_0	b_1	b_2	b_3
Δx	1.687	1.707e2	2.824e2	8.673e4
Δy	1.439	2.404e2	3.448e2	9.977e4

Table 9: Lower Bound and Upper Bound for Elastic Modulus Values for Coarse Model 2

	Lower Bound	Upper Bound
E	5.562e4	5.573e4

The resulting P-Box distribution of the coarse scale model as compared to the fine scale model can be seen in Figure 5-30.

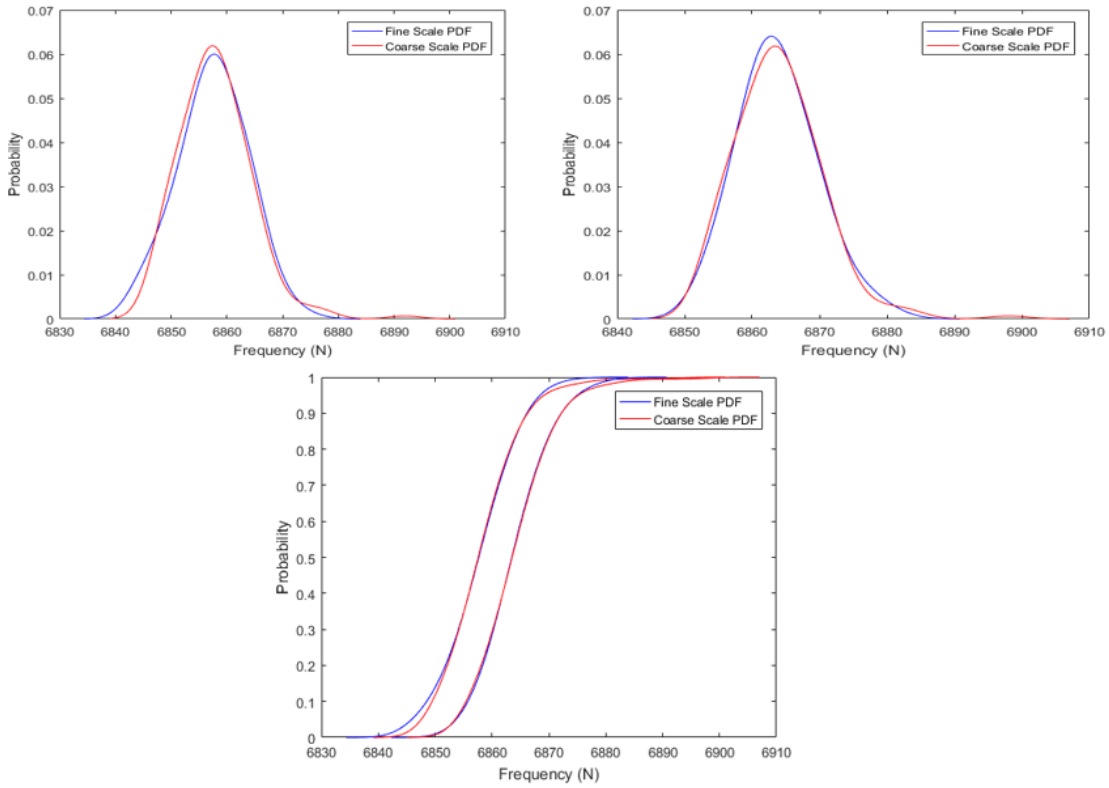


Figure 5-30: Upscaling Results for Coarse Model 2 - a) Left Side PDF b) Right Side PDF c) P-Box

Both coarse scale models appear to give good results for matching the P-Box of the first natural frequency.

5.3 Validation of Results

Using the validation methods discussed in section 2.8, the coarse scale models developed were compared to experimental results. This section will cover the experimental setup as well as the results of the validation.

5.3.1 Experimental Setup

In order to test the natural frequency of the turbine blade, an experiment was designed to find the natural frequency of the manufactured specimen. A layout of this experiment can be seen in Figure 5-31.

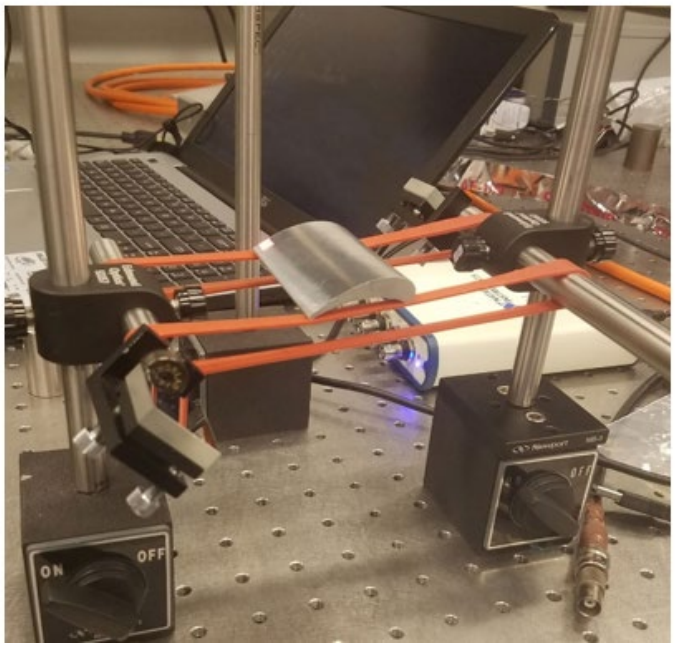


Figure 5-31: Experimental Setup

In this experiment, the turbine blade is suspended using rubber bands. This represents the free-free boundary condition used in the simulations. Because the rubber bands have a resonance frequency much lower than the turbine blade, it has no effect on recording the natural frequencies of the turbine blade. The turbine blade was excited using a hammer with a force transducer attached. The force transducer records the input into the system (the turbine blade). The response of the turbine blade was recorded using a laser doppler vibrometer (LDV). By taking the transfer function of the input (force transducer) over the output (LDV), a transfer function can be found. This is done using equation 24.

$$TF = 20\log\left(\frac{Output}{Input}\right) \tag{24}$$

This process is repeated several times and the average is taken to get a final transfer function. An example of the transfer function can be seen in Figure 5-32.

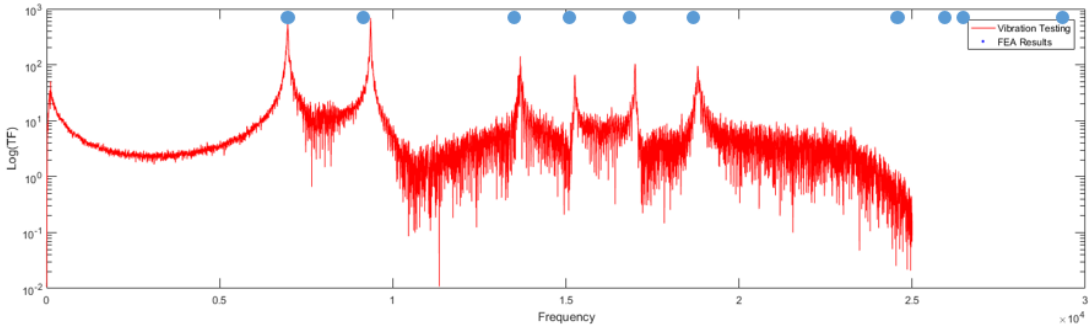


Figure 5-32: Example Transfer Function with Experimental and Average Simulation Results

The blue dots represent the average value of the natural frequencies in the simulation. Since the deterministic results can be misleading, validation was conducted with the stochastic

responses. This process was repeated for 3 blades. The first natural frequency for each of the blades can be seen in Table 10.

Table 10: Natural Frequencies of Manufactured Turbine Blades

6835 Hz
6894 Hz
6925 Hz

5.3.2 Validation Metrics

Validation was completed separately for both aleatory and p-box responses. For the aleatory response obtained in section 5.1, K-S test was used as the validation metric. For the p-box response obtained in section 5.2, the mean curve based method was used since very little experimental samples were available.

5.3.2.1 Aleatory Response

The K-S test was used to validate the aleatory response. A graph of the empirical cumulative distribution function (ECDF) as compared to the aleatory simulation results can be seen in Figure 5-33. The ECDF is shown in blue, the simulation CDF is shown in orange, and the maximum distance between the two is shown in green. This distance is plugged into the K-S distribution to yield a p value of 0.0800. This means that the null hypothesis that the two distributions are sampled from the same distribution is not rejected at a significance value of 0.05.

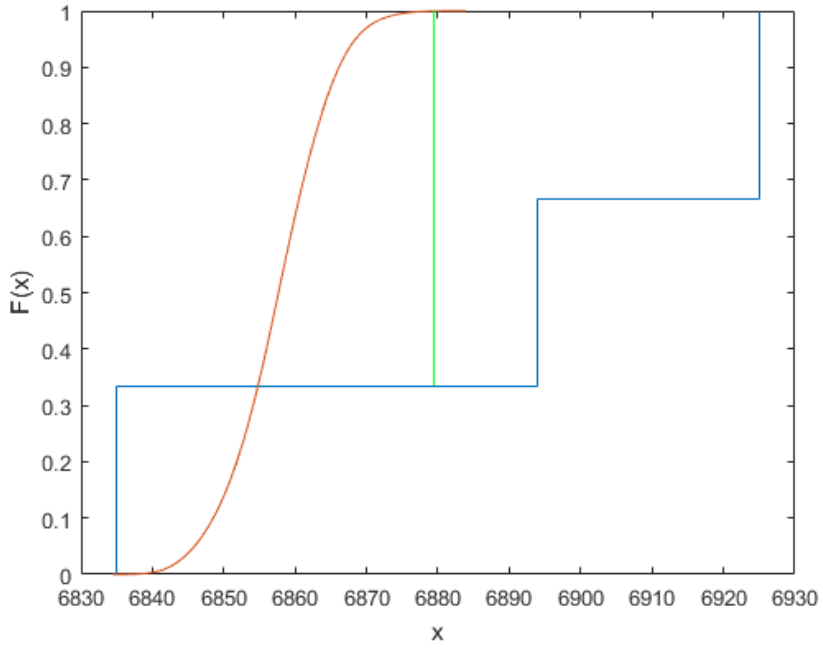


Figure 5-33: K-S Test for Aleatory Response

5.3.2.2 P-Box Response

The mean curve method was used to validate the P-Box response of the simulation results. The mean curve (shown in green), along with the P-Box and ECDF of the experimental results can be seen in Figure 5-34. Using the K-S test with the ECDF of the experimental results and the mean curve yields a value of 0.0800, meaning that the null hypothesis that the two distributions are sampled from the same distribution is not rejected at a significance value of 0.05.

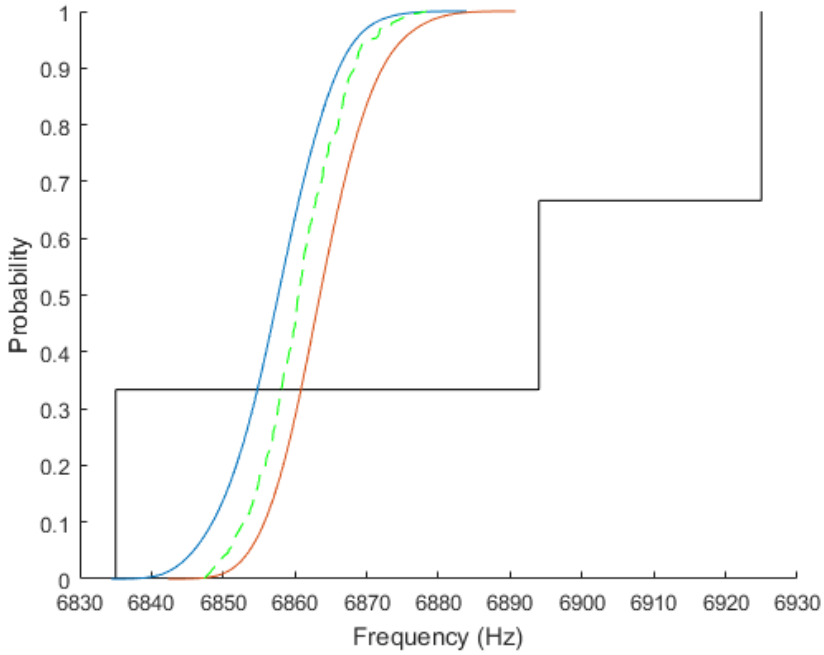


Figure 5-34: Mean Curve Validation Method for P-Box Response

CHAPTER 6. CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this project, an upscaling and validation framework was developed in order to efficiently quantify and integrate geometric and material uncertainty into simulation using a turbine blade example. First, the geometric uncertainty of a machined aluminum turbine blade was quantified using a CMM machine. Using semivariogram analysis and K-L expansion, a finite element model of a turbine blade was modified, and stochastic simulations were conducted to obtain a distribution for the first natural frequency. A coarse scale model was generated using Bezier curves to model the geometry and Delaunay triangulation to mesh the turbine blade. Finally, a stochastic upscaling method was applied to the coarse scale model to match the first 2 natural frequencies.

In the material UQ section, the upscaling process was used to obtain a distribution of elastic moduli. First, a CT scanner was used to analyze the porosity of the material used to manufacture the turbine blade. The CT scan images were converted to voxel models of the pores inside of the material. The size of the pores were quantified using Bayesian Information Criterion and fine scale beam simulations were conducted to obtain distributions of reaction force when strain was applied. A simple coarse model was created, and the stochastic upscaling method was used to find a distribution of elastic moduli such that the behavior of the coarse scale model matched that of the fine scale models. The elastic moduli were applied to the fine scale model created in Section 5.1, and stochastic simulations with both epistemic and aleatory uncertainty were conducted to obtain a P-Box for the first natural frequency. 2 candidate coarse scale models were then created and

stochastic upscaling was used to match the first natural frequency P-Box to that of the fine scale model. Finally, experiments were carried out and the results were validated using validation metrics for aleatory and epistemic uncertainty.

The validation results show that the aleatory and P-Box responses for first natural frequency are validated using the 3 samples that were tested. Due to the low number of samples and primitive experimental setup, this experimental validation could be greatly improved in the future. Using the turbine blade example, the framework was able to successfully quantify and propagate both geometric (taken to aleatory uncertainty) and material (taken to be epistemic uncertainty) variation into simulation and use the stochastic upscaling process to obtain coarse scale models that match the result. These coarse scale models have near identical responses to the fine scale models while taking a fraction of the time to perform simulations. A single fine scale simulations takes about 5 minutes to perform while the coarse scale simulations only take approximately 20 seconds to perform. This increase in computational efficiency is useful when performing many simulations.

6.2 Future Work

As mentioned earlier, an improvement to the experimental setup as well as an increase in the number of samples could greatly increase the confidence in validating the model or show that the model is not validated. In addition, applications of the obtained coarse scale models could be explored. Reliability analysis is an application that often requires hundreds of thousands of simulations to accurately represent high reliability situations. In addition, if a different input variable (other than scaling factor) was used for the upscaling process, the coarse scale models of the turbine blades could be substituted

into models for entire turbine engines, to greatly reduce the complexity of the simulations. The efficacy and effectiveness of this process could be explored in the future. In addition, upscaling with multiple outputs was attempted, but only for predicting different natural frequencies. It would be notable to attempt to use multi-objective stochastic upscaling to match two different outputs, such as first natural frequency and maximum tip deflection.

Since this framework was created for any generic part, a software capable of applying this method to any type of part would be extremely useful. Production of this software has been started using Python GUI libraries, but much could be done to improve the design and capabilities of the software. The software is broken down into 4 different sections: Geometric UQ Toolbox, Material UQ Toolbox, Upscaling Toolbox, and Validation Toolbox. The purpose of the Geometric UQ Toolbox is to take in data from the CMM machine and generate semivariogram data. The purpose of the Material UQ Toolbox is to take in data from a CT scanner and use the proposed methods in Chapter 4 to generate a P-Box for the elastic modulus. The purpose of the Upscaling Toolbox is to use the generated semivariogram and elastic modulus data to handle fine scale simulations and use stochastic upscaling to create a coarse scale model that matches the outputs of the coarse scale model to those generated by the fine scale model. A more in depth description of each section as well as some images from the current progress of the developed software can be seen in Appendix A.

APPENDIX A. DEVELOPED SOFTWARE FOR FRAMEWORK

A.1 Geometric UQ Toolbox GUI

The geometric UQ toolbox simply has 1 section. The purpose is to take in CMM data and create a data structure containing the semivariogram information. The layout of the geometric UQ toolbox can be seen in Figure A-1.

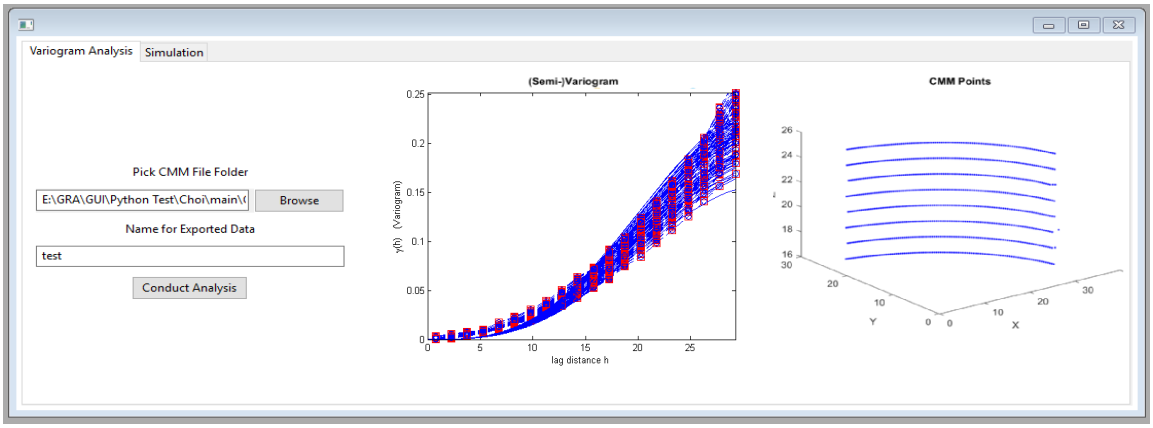


Figure A-1: Geometic UQ Toolox GUI

The required inputs are the folder containing the CMM data, and a name for the exported data. After clicking the "Conduct Analysis" button, a separate variogram analysis is completed for each section of CMM data. In addition, a 3D graph of the points that were analyzed is shown in order to give a visual cue to the user. The data is saved and used later in the upscaling toolbox. This section is designed so that the variogram analysis can be separated by section. Such as in the turbine blade example, the blade was separated into 4 sections: concave, convex, small edge, and large edge. These would be performed separately and saved as different files.

A.2 Material UQ Toolbox GUI

The purpose of the material UQ toolbox is to take in CT scan data and create a p-box for elastic modulus. This sections is split into two different tabs: A CT scan analysis tab and a fine scale simulation tab. The CT scan analysis tab can be seen in Figure A-2.

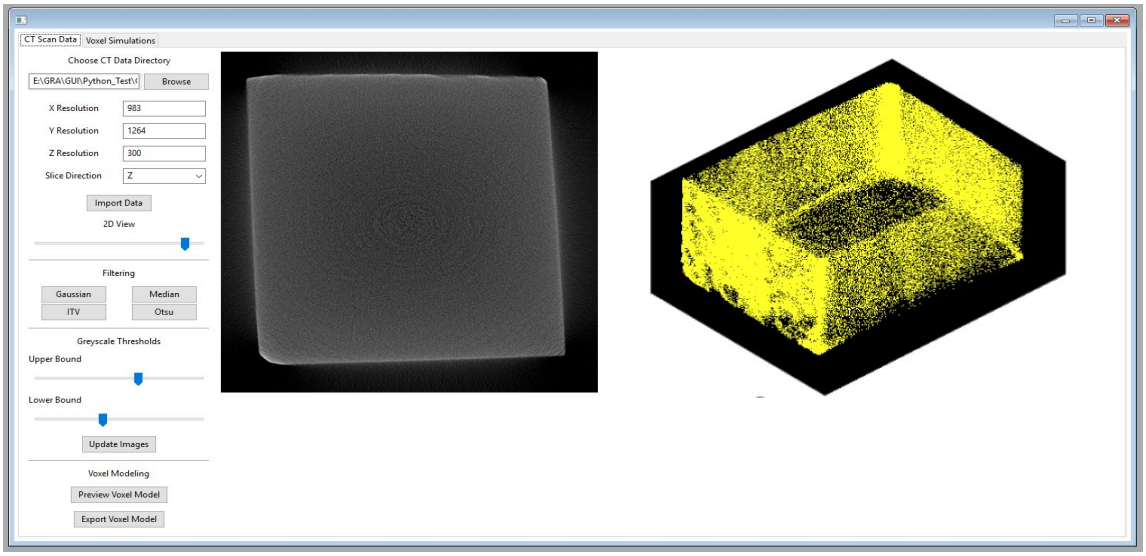


Figure A-2: Geometry UQ Toolbox

The inputs that are required is the folder containing the CT scan data, the resolution of the data, and a chosen slice direction. After clicking the "Import Data" button, the CT scan data is imported and sliced into 2D pictures according to the chosen slice direction. Once the data is finished importing, the 2D slices are shown to the right of input section. The 2D slices can be navigated using the slider below the input section. In addition, options are given below to set black/white cutoff values as well as perform filtering. In the turbine example, Gaussian filtering was used to filter the CT scan data, but other filtering options such as median, ITV, and Otsu are given as options as well. Below the filtering section are

greyscale threshold sliders. These sliders set the maximum and minimum greyscale values allowed in the image. This can help better identify the pores located in the material.

After completing the filtering and greyscale threshold processes, the voxel model can then be exported. Clicking the “Preview Voxel Model” button, will show a graph of what the final voxel model will look like on the right. This is used to help the user get an idea of the final voxel model before exporting, as exporting can take a long time. Clicking the “Export Voxel Model” button will export the determined voxel model as an Abaqus input file.

The second tab of the material UQ toolbox is designed to take the exported voxel models from the first tab and perform the beam voxel fine scale simulations to find a p-box distribution of homogenized elastic modulus. This section is still currently being constructed as the focus has been on improving the speed of the first tab.

A.3 Upscaling Toolbox GUI

The purpose of the upscaling toolbox is designed to handle all of the fine and coarse scale simulations, as well as perform the upscaling process to match the responses. The upscaling toolbox is split into 4 different tabs: Fine Scale Simulations (Aleatory), Upscaling (Aleatory), Fine Scale Simulations (P-Box) and Upscaling (P-Box). The fine scale simulation and upscaling processes are vastly different when epistemic uncertainty is introduced, so these situations are separated.

The Fine Scale Simulations (Aleatory) tab can be seen in Figure A-3.

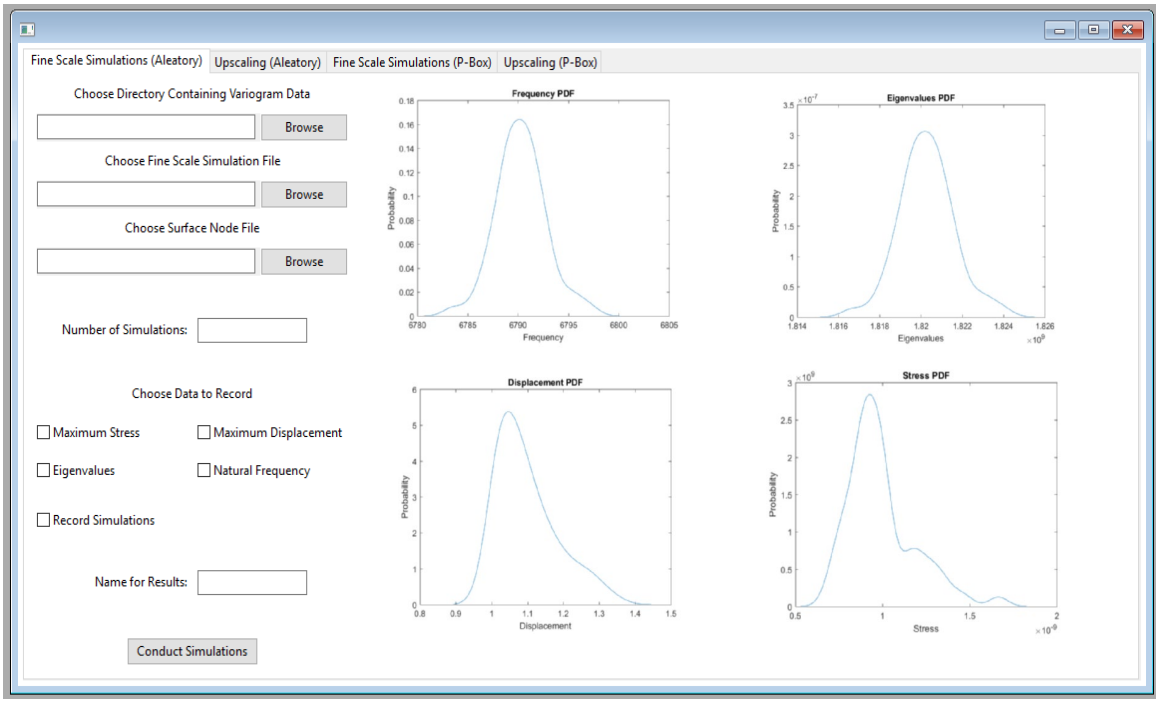


Figure A-3: Fine Scale Simulations (Aleatory) Tab of Upscaling Toolbox

The inputs required are the semivariogram data from the geometry UQ toolbox, a fine scale simulation file (in the form of an Abaqus input file), and a text file containing a list of all of the surface nodes in the fine scale simulation. With this information it is possible to use the semivariogram data to modify the surface coordinates in the Abaqus input file to conduct stochastic simulations. Next, the number of simulations as well as the required output data are chosen. Choosing different combinations of output data will construct a different python file for reading the Abaqus output file. This way, it does not look for results that do not exist in the output file. Finally, a name for the results can be chosen. After clicking the "Conduct Simulations" button, the fine scale simulations begin. Since the process can take a very long time, a progress bar was introduced to get an idea of how

much time remains in the simulation. After the simulations are done, graphs of the results are shown on the right side, depending on what was chosen.

The Upscaling (Aleatory) tab can be seen in Figure A-4.

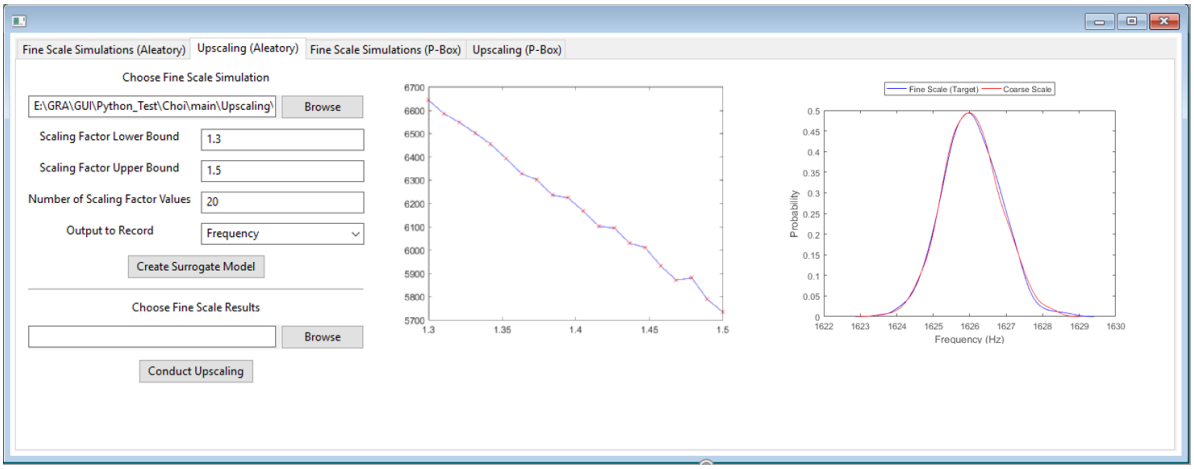


Figure A-4: Upscaling (Aleatory) Tab of Upscaling Toolbox

For this tab, the original fine scale simulation file is needed. This is used to create a coarse scale representation of the turbine blade using the Bezier curve approximation method. In addition, the lower bound, upper bound, and number of scaling factor values that are desired in the Kriging operation must be chosen. After clicking the "Create Surrogate Model" button, the coarse scale model is constructed and simulated at values in between the lower and upper bound of the scaling factor. In addition, the desired output to be modeled must be chosen. The choices are the same as those in the Fine Scale Simulations (Aleatory) tab, but only one can be chosen for Kriging. The points as well as the resulting Kriging model are shown to the right of the input. Finally, the fine scale results must be chosen. After clicking the "Conduct Upscaling" button, the scaling factor distribution is

determined so that the response of the coarse scale model matches that of the fine scale model. The final coarse and fine scale distributions are shown to the right.

The Fine Scale Simulations (P-Box) tab is shown in Figure A-5.

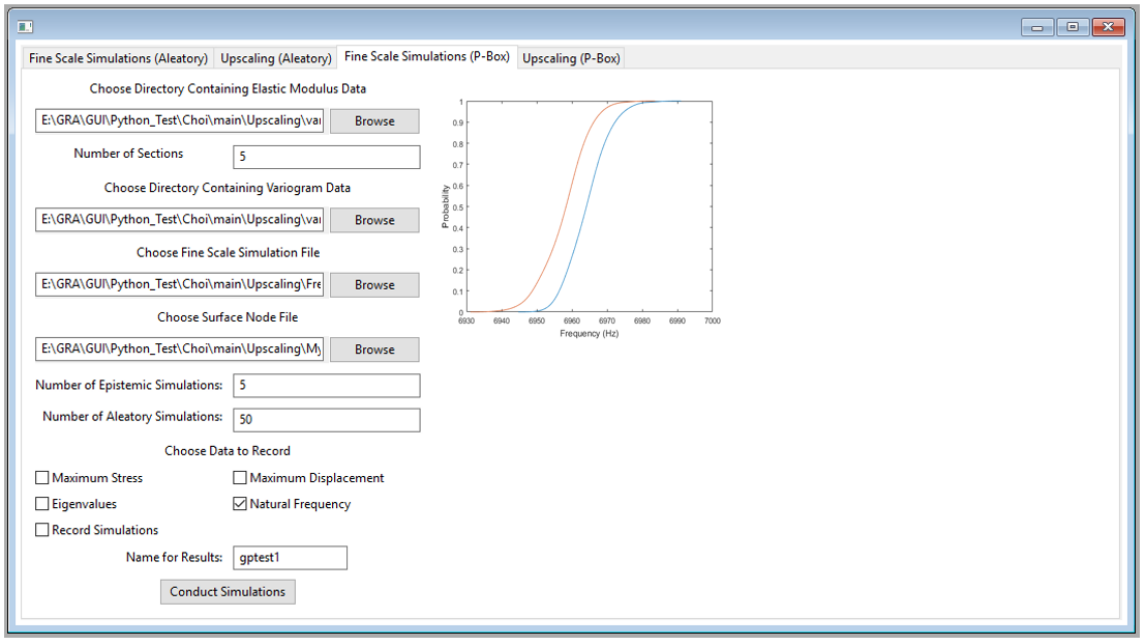


Figure A-5: Fine Scale Simulations (P-Box) Tab of Upscaling Toolbox

The layout of this tab is similar to the Fine Scale Simulations (Aleatory) tab, but in addition requires information on the elastic modulus data that was obtained from the material UQ toolbox. Also additional information on how many simulations are performed with varying epistemic vs varying aleatory uncertainty is required. Once again, any combination of outputs can be chosen. The results will appear to the right of the input.

A.4 Validation Toolbox GUI

The validation toolbox is split into 3 different tabs depending on the type of validation being used: Single Site Validation, Multiple Site Validation, and P-Box validation. The layout of the single site validation tab can be seen in Figure A-6.

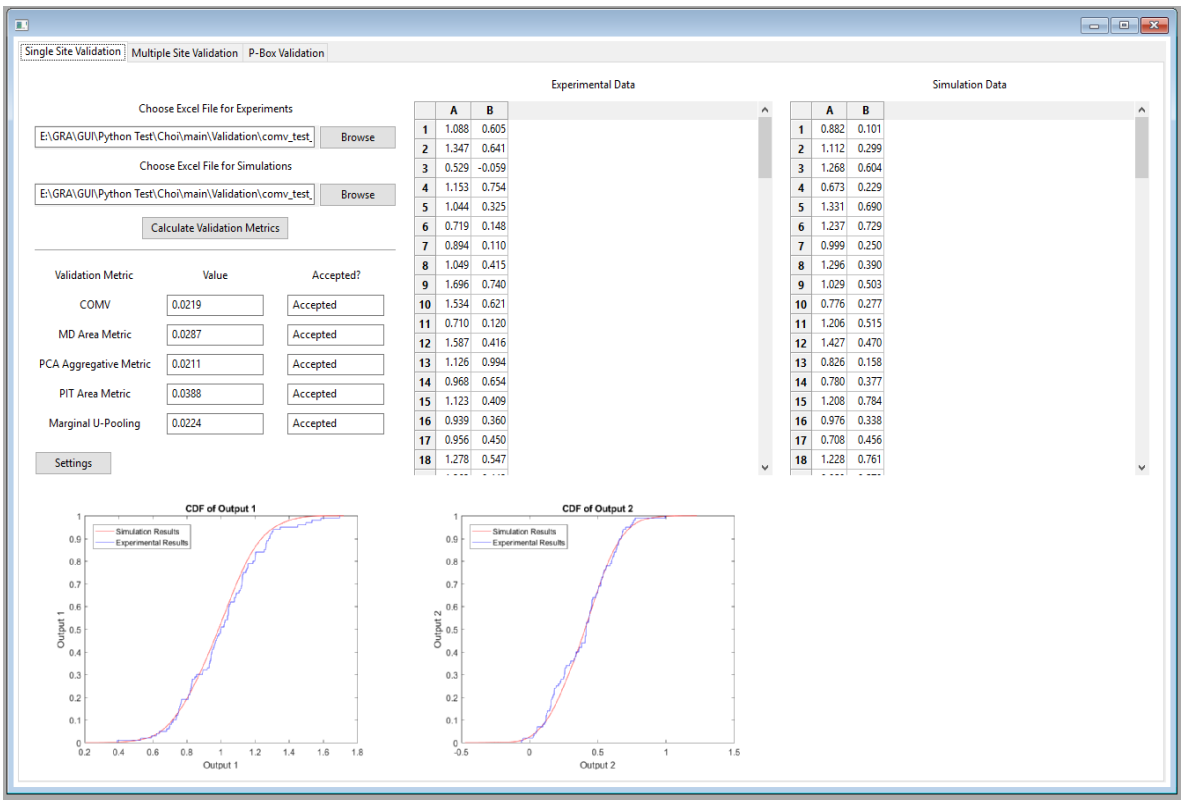


Figure A-6: Single Site Validation Tab of Validaiton Toolbox

As an input, 2 excel sheets are required. The first excel sheet must contain the results from the experiments, and the second excel sheet must contain the results from the simulations. As many outputs as desired can be validated simply by including the data as extra columns. After clicking the "Calculate Validation Metrics" button, various validation metrics will be calculated for single site validation as well as whether or not they are

determined to accept the results. In addition, graphs of the experimental and simulation results for each output are shown in the space below.

The multiple site validation tab has a similar layout, and is shown in Figure A-7.

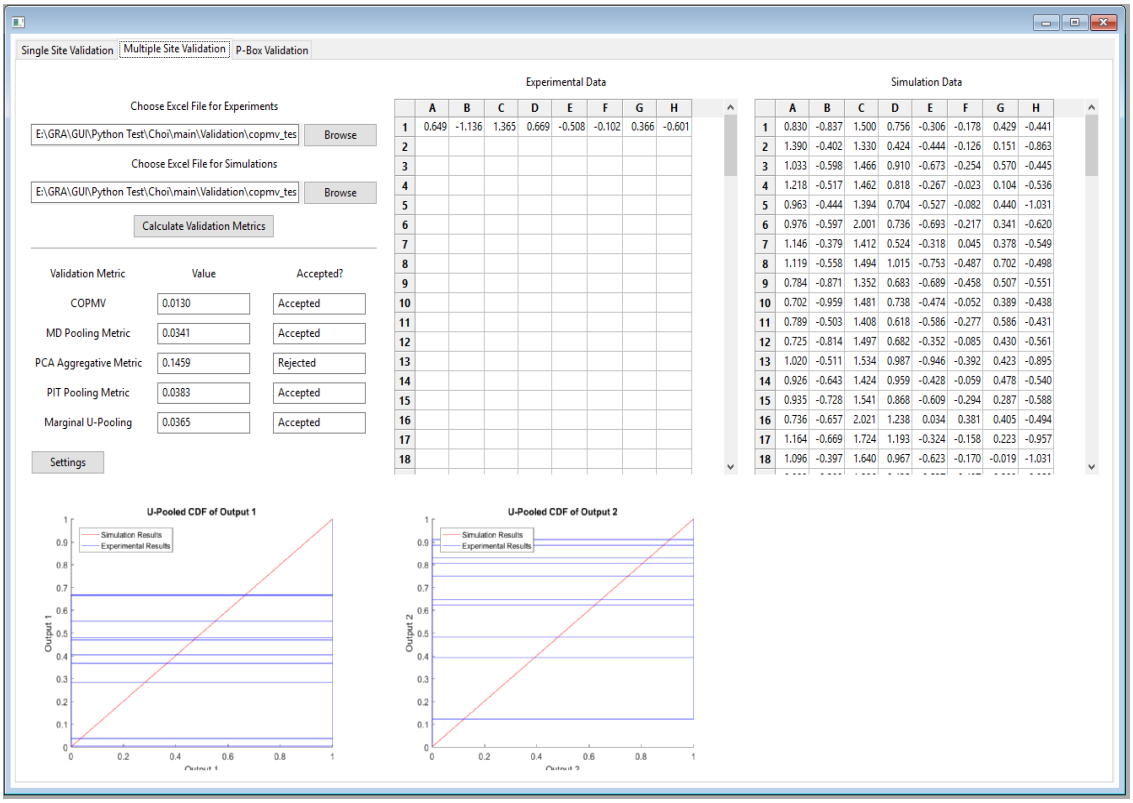


Figure A-7: Multiple Site Validation Tab of Validation Toolbox

As before in the single site validation tabs, two excel sheets are needed as the input. After clicking the "Calculate Validation Metrics" button, various validation metrics are calculated and graphs of the results are shown below.

The P-box validation tab is also laid out in a similar way, with the addition of a third input option containing resimulated data. The layout of the P-box validation tab can be seen in Figure A-8.

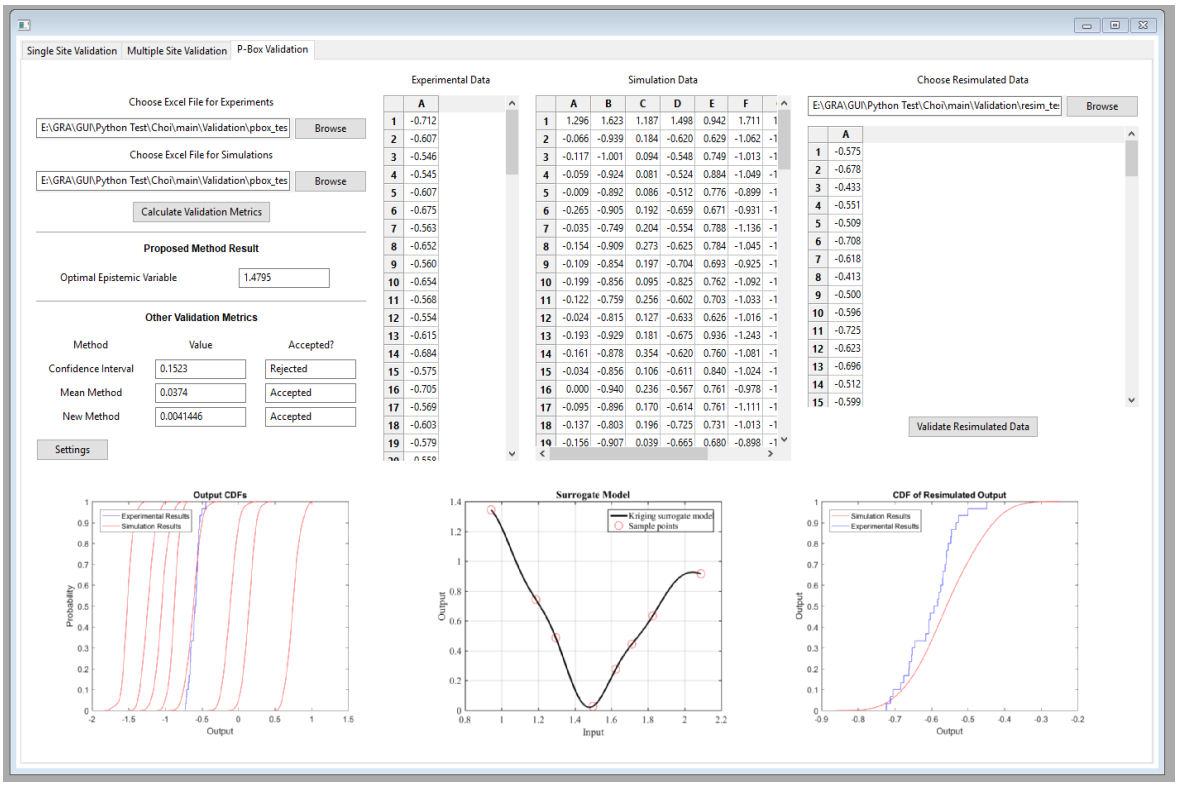


Figure A-8: P-Box Validation Tab of Validation Toolbox

Once again, an excel file for the experiments and an excel file for the simulation results are required as inputs. After clicking the "Calculate Validation Metrics" button, the mean method and confidence interval method are calculated, as well as the optimal epistemic value in the new proposed method. A graph of simulation and experimental results as well the developed kriging model are shown below. After resimulating with the optimal epistemic value, the new results can be chosen as a third input. Clicking the "Validate Resimulated Data" will validate using the new method and show the new resimulated data below.

REFERENCES

[1]. Hou, J., Wicks, B. J., Antoniou, R. A., "An Investigation of Fatigue Failures of Trubine Blades in a Gas Turbine Engine by Mechanical Analysis," Engineering Failure Analysis **9** pp. 201-211 (2002).

[2]. Mazur, Z., Luna-Ramirez, A., Juarez-Islas, J. A., Campos-Amezcua, A. "Failure Analysis of a Gas Turbine Blade Made of Inconel 738LC Alloy," Engineering Failure Analysis **12** pp. 474-486 (2005).

[3]. Garzon, V.E. and Darmofal, D.L., 2003, January. Impact of geometric variability on axial compressor performance. In *ASME Turbo Expo 2003, collocated with the 2003 International Joint Power Generation Conference*, pp. 1199-1213.

[4]. Carnevale, M., Montomoli, F., D'Ammaro, A., Salvadori, S. and Martelli, F., 2013. "Uncertainty quantification: A stochastic method for heat transfer prediction using LES." Journal of Turbomachinery, 135(5), p.051021.

[5]. Javed, A., Pecnik, R. and Van Buijtenen, J.P., 2016. "Optimization of a Centrifugal Compressor Impeller for Robustness to Manufacturing Uncertainties." Journal of Engineering for Gas Turbines and Power, **138**(11), p.112101.

[6]. Lange, A., Voigt, M., Vogeler, K., Schrapp, H., Johann, E. and Gümmer, V., 2012. Impact of manufacturing variability on multistage high-pressure compressor performance. Journal of Engineering for Gas Turbines and Power, 134(11), p.112601.

[7]. Thakur, N., Keane, A.J., Nair, P.B. and Rao, A.R., 2010. "Probabilistic life assessment of gas turbine blades." Journal of Mechanical Design, 132(12), pp.121005.

[8]. Nagpal, V. K., Rubinstein, R., Chamis, C. C., 1989, "Probablisitic Structural Analysis to Quantify Uncertainties Associated with Turbopump Blades." American Institute of Aeronautics and Astronautics Journal, **27**(6).

[9]. Zhu, S. P., Liu, Q., Zhou, J., Yu, Z. Y., 2017, "Fatigue Reliability Assesssment fo Turbine Discs Under Multi-Source Uncertainties." Fatigue and Fracture of Engineering Materials and Structures **41** pp. 1291-1305.

[10]. Weiss, T., Voigt, M., Schlums, H., Mucke, R., Becker, K. H., Vogeler, K. 2009. "Probabilistic Finite-Element Analyses on Turbine Blades." *ASME Turbo Expo: Power for Land, Sea and Air.*

[11]. Shen, M. H., 1999. “Reliability Assessment of High Cycle Fatigue Design of Gas Turbine Blades Using the Probabilistic Goodman Diagram.” International Journal of Fatigue, **21** pp. 699-708.

[12]. Choi, S., Grandhi, R.V., and Canfield, R.A., 2006, Reliability-Based Structural Design, Springer, London.

[13]. Ross, J. L., Ozbek, M. M., Pinder, G. F., 2009. “Aleatoric and Epistemic Uncertainty in Groundwater Flow and Transport Simulation.” Water Resources Research, 45, W00B15.

[14]. Hofer, E., Kloos, M., Krzykacz-Hausmann, J. P., Woltereck, M., 2002. “An Approximate Epistemic Uncertainty Analysis Approach in the Presence of Epistemic and Aleatory Uncertainties.” Reliability Engineering and System Safety, 77 pp. 229-238,

[15]. Oberkampf W. L., DeLand, S. M., Rutherford, B. M., Diegert, K. V., Alvin, K. F., 2002. “Error and Uncertainty in Modeling and Simulation.” Reliability Engineering and System Safety, 75 pp. 333-357.

[16]. Schobi, R., Sudret, B., “Uncertainty Propagation of P-Boxes Using Sparse Polynomial Chaos Expansions,” *Chair of Risk, Safety and Uncertainty Quantification, ETH Zurich, Stegano-Franscini-Platz 5, 8093 Zurich, Switzerland* (2016).

[17]. Karanki, D. R., Kushwaha, H. S., Verma, A. K., Ajit, S., 2009. “Uncertainty Analysis Based on Probability Bounds (P-Box) Approach in Probabilistic Safety Assessment.” Risk Analysis, 29(5).

[18]. Yvonnet, J., Monteiro, E. and He, Q.C., 2013. Computational Homogenization Method and Reduced Database Model for Hyperelastic Heterogeneous Structures. International Journal for Multiscale Computational Engineering, **11**(3). pp. 201-225.

[19]. Allen, J.K., Seepersad, C., Choi, H., and Mistree, F., 2006. Robust Design for Multiscale and Multidisciplinary Applications. Journal of Mechanical Design, **128**(4), pp. 832-843.

[20]. Choi, H., McDowell, D.L., Allen, J.K., Rosen, D., and Mistree, F., 2008. An Inductive Design Exploration Method for Robust Multiscale Materials Design. Journal of Mechanical Design, **130**(3), pp. 031402.

[21]. Chen, W., Yin, X., Lee, S., and Liu, W.K., 2010. A Multiscale Design Methodology for Hierarchical Systems with Random Field Uncertainty. Journal of Mechanical Design, **132**(4), pp. 041006.

[22]. Ruderman, A., Choi, S.K., Patel, J., Kumar, A., and Allen, J.K., 2010. Simulation Based Robust Design of Multiscale Products. Journal of Mechanical Design, **132**(10), pp. 101003.

[23]. ASME Committee PTC-60 V&V 10, 2006. Guide for Verification and Validation in Computational Solid Mechanics, ASME.

[24]. Post D. E., The Coming Crisis in Computational Sciences, 2004. Technical Report Rep.LA-UR-03-88-2004, Los Alamos National Laboratory, Los Alamos, NM.

[25]. Liu, Y., Chen, W., Arendt, P., and Huang, H. Z., 2011. "Toward a Better Understanding of Model Validation Metrics," Journal of Mechanical Design,**133**(7), pp. 071005.

[26]. Curran, P.J., 1988. "The semivariogram in remote sensing: an introduction," Remote sensing of Environment, **24**(3), pp.493-507.

[27]. Jian, X., Olea, R.A. and Yu, Y.S., 1996. "Semivariogram modeling by weighted least squares," Computers & Geosciences, **22**(4), pp.387-397.

[28]. Loeve, M., 1978, Probability Theory II, Springer-Verlag, New York.

[29]. Ghanem, R. G., Spanos, P. D., 1991, Stochastic Finite Element Analysis: A Spectral Approach, Springer-Verlag, New York.

[30]. Wiener, N., 1938. The Homogeneous Chaos. American Journal of Mathematics, 60(4), pp.897-936.

[31]. Gorguluarslan, R. M., "A Multi-level Upscaling and Validation Framework for Uncertainty Quantification in Additively Manufactured Lattice Structures," Georgia Institute of Technology, Atlanta, GA, (2016).

[32]. Gorguluarslan, R. and Choi, S.K., 2014. A simulation-based upscaling technique for multiscale modeling of engineering systems under uncertainty. International Journal for Multiscale Computational Engineering, 12(6).

[33]. Gorguluarslan, R.M., Park, S.I., Rosen, D.W., and Choi, S.-K., 2015, "A Multilevel Upscaling Method for Material Characterization of Additively Manufactured Part under Uncertainties." Journal of Mechanical Design, 137(11), pp. 111701.

[34]. Davidoiu, V., Hadjilucas, L., Teh, I., Smith, N. P., Schnieder, J. E., Lee, J., 2016. "Evaluation of Noise Removal Algortihms for Imaging and Reconstruction of Vascular Networks Using Micro-CT." Biomedical Physics and Engineering Express, 2, 045015.

[35]. Shirani, M., Harkegard, G., 2012. “Damage Tolerant Design of Cast Components based on defects detected by 3D X-ray Computed Tomography.” International Journal of Fatigue, **41** pp. 188-198.

[36]. Shin, D. H., Park, R. H., Yang, S. J., Jung, J. H., 2005. “Block-Based Noise Estimation Using Adaptive Gaussian Filtering.” IEEE Transaction on Consumer Electronics, 51(1).

[37]. Noh, Y., Choi, K.K. and Lee, I., 2010. “Identification of Marginal and Joint CDFs Using Bayesian Method for RBDO,” Structural and Multidisciplinary Optimization, **40**(1-6), pp.35-51.

[38]. Posada, D., and Buckley, T.R., 2004.” Model Selection and Model Averaging in Phylogenetics: Advantages of Akaike Information Criterion and Bayesian Approaches over Likelihood Tests,” Systematic Biology, **53**, pp. 793-808.

[39]. Hurvich, C.M. and Tsai, C.L., 1989. “Regression and Time Series Model Selection in Small Samples,” Biometrika, **76**(2), pp.297-307.

[40]. Beck, J.L. and Yuen, K.V., 2004. “Model Selection Using Response Measurements: Bayesian Probabilistic Approach,” Journal of Engineering Mechanics, **130**(2), pp.192-203.

[41]. Oberkampf, W.L., C. Roy, 2010. “Verification and Validation in Scientific Computing,” Cambridge University Press, Cambridge, UK.

[42]. Ling, Y., and Mahadevan, S., 2013. “Quantitative Model Validation Techniques: New Insights,” Reliability Engineering and System Safety, Vol. 111, pp. 217-231.

[43]. Oberkampf, W. L., Trucano, T. G., and Hirsch, C., 2004. Verification, Validation, and Predictive Capability in Computational Engineering and Physics. Applied Mechanics Reviews, 57(3), pp. 345-384.

[44]. Feiyun, C., Chen, J., Pan, Y. “Kolmogorov-Smirnov Test for Rolling Bearing Performance Degradation Assessment and Prognosis,” Jounal of Vibration and Control **17**(9) pp. 1337-1347 (2010).

[45]. Sebastien D., Olivier S.. “Kolmogorov-Smirnov Test for Interval Data.” *15th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, Communications in Computer and Information (444), pp.416-425, 2014.

[46]. Voyles, I. T., Roy, C. J., “Evaluation of Model Validation Techiniques in the Presence of Aleatory and Epistemic Input Uncertainties,” *AIAA Science and Techonlogy Forum 2015.*

[47]. Ferson, S., Oberkampf, W. L., Ginzburg, L., “Validation of Imprecise Probability Models,” Applied Biomathematics (2008).

[48]. Samareh, J.A., 2001. Survey of shape parameterization techniques for high-fidelity multidisciplinary shape optimization. AIAA journal, 39(5), pp.877-884.

[49]. Mohaghegh, K., Sadeghi, M.H. and Abdullah, A., 2007. "Reverse engineering of turbine blades based on design intent." The International Journal of Advanced Manufacturing Technology, 32(9-10), pp.1009-1020.

[50]. Nemnem, A.F., Turner, M.G., Siddappaji, K. and Galbraith, M., 2014, June. A smooth curvature-defined meanline section option for a general turbomachinery geometry generator. In ASME Turbo Expo 2014: Turbine Technical Conference and Exposition (pp. V02BT39A026-V02BT39A026).

[51]. Vucina, D., Lozina, Z. and Pehnec, I., 2008, July. "A compact parameterization for shape optimization of aerofoils." In Proceedings of the World Congress on Engineering (Vol. 1, pp. 111-116).

[52]. Farin, G., 1993, Curves and Surfaces for Computer Aided Geometric Design, 2nd Ed., Academic Press: New York.

[53]. Pérez-Arribas, F. and Trejo-Vargas, I., 2012. "Computer-aided design of horizontal axis turbine blades." Renewable energy, 44, pp.252-260.

[54]. Giovannini, M., Marconcini, M., Rubechini, F., Arnone, A. and Bertini, F., 2016. "Scaling Three-Dimensional Low-Pressure Turbine Blades for Low-Speed Testing. Journal of Turbomachinery", 138(11), p.111001.

[55]. Lee, K., 1999, Principles of CAD/CAM/CAE Systems, Addison Wesley.

[56]. Lee, D. T., and Schachter B. J., 1980, "Two Algorithms for Constructing a Delaunay Triangulation. International Journal of Computer and Information Sciences," 9(3). Pp. 219-242.

[57]. Shewchuk, and Jonathan R., 2002, "Delaunay Refinement Algorithms for Triangular Mesh Generation." Computational Geometry, 22, pp. 21–74.

[58]. Reddy, J. N., *An Introduction to the Finite Element Method*. McGraw Hill. New York, NY, (2006).

[59]. GrabCAD, Turbine blade from a Rolls Royce Pegasus turbofan engine. Retrieved from: https://grabcad.com/library/turbine-blade--12

[60]. Becker, Andrew, and Marsden, Paul, 1999, "Frequency Screening of First Stage Turbine Blades in the T56-A-7B Engine." Airframes and Engines Division Aeronautical and Maritime Research Laboratory. DSTO-TN-0223.

[61]. Hang S. "TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator," ACM Trans. on Mathematical Software. **41** (2), Article 11 (February 2015).

[62]. Cressie, N. “The Origins of Kriging,” Mathematical Geology **22** pp. 239-252.

[63]. Lophaven, Soren N., Nielsen, Hans B., and Sondergaard, J. 2002, “A MATLAB Kriging Toolbox.” Technical University of Denmark.